Fermilab

User's Guide to ESME 2001

James A. MacLachlan   Jean-François Ostiguy

September 8, 2000

Abstract

ESME is a computer program to calculate the evolution of a distribution of particles in energy and azimuth as it is acted upon by the radio frequency system of a synchrotron or storage ring. It provides for the modeling of multiple rf systems, feedback control, space charge, and many of the effects of longitudinal coupling impedance.

The capabilities of the program are described, and the requirements for input data are specified in sufficient detail to permit significant calculations by an uninitiated user. This manual describes ESME 2001 , extensively modified since the previous user documentation. The principal changes in input parameters since v. 7.1 (1987) are tabulated.

Optional command line arguments control the volume of output, interactive mode, etc. Memory allocation can be controlled at run time for some large arrays.

Contents

Chapter 1

Introduction

The program ESME has been developed to model those aspects of beam behavior in a synchrotron that are governed by the radio frequency systems. It follows the evolution of a distribution in energy-azimuth coordinates by iterating a map corresponding to the single-particle equations of motion. A provision for scaling the map for faster calculations has been introduced in the 2001 version;[1] only when the scaling constant is one does an iteration correspond to a beam turn. The map parameters may be updated each iteration to reflect the action of the beam current on the individual particles through feedback loops, space charge, coupling impedance, *etc*. The code was initially developed during the years 1981–82 for the design of the Tevatron I Antiproton Source[1] and documented for general use in 1984.[5] This manual documents a substantial revision of the program. The previous supported version was ESME v. 8.5. Data prepared for this and earlier versions are likely to need some modification to run under ESME 2001. In Appendix A is given a table of obsolete input parameters and their v. 2001 equivalents and the parameters first appearing in ESME2000. The conceptual basis of ESME is discussed at length elsewhere.[6]

ESME is used frequently to assess the efficiency of a given rf beam manipulation or to optimize system parameters. For this the user needs to specify technical details of the various subsystems and derive various numerical measures of system performance from the particle distribution. In this sort of calculation many data are required, and the program may employ several numerical analysis features. Equally useful, however, are qualitative calculations designed to illustrate a concept or explore the feasibility of a novel approach. For such use the code requires a minimum of data and provides easy access to a variety of graphical output. When a preliminary investigation has been fruitful it is natural to proceed to a thorough modeling of the process. ESME is intended to serve effectively over a wide range of detail by separating functions so that only the relevant ones need be considered and by establishing reasonable defaults to reduce the data required for typical cases. Thus, a few lines of data may serve to get a first look at a system which can be studied in greater detail by overriding defaults with specific input and by using additional functions like, for example, those related to collective behavior or those related to numerical evaluation of the properties of the distribution.

1.1   Coordinate System

The macroparticle coordinates in ESME are energy difference [MeV] from the synchronous energy and phase difference from the synchronous phase. Because the program accommodates simultaneous operation of rf systems at different harmonic number h, the phase variable is taken at $h = 1$. This means, for example, that the unstable fixed points for a stationary bucket occur at $\pm \pi/h$. Internally the phases are in radians, but input data are in degrees. The single-particle longitudinal dynamics depend on the mean radius of the ring $R_{\mathrm{eq}} = C/2\pi$, the momentum $p_\circ$ (energy $E_\circ$) of a particle which follows the central orbit, and the momentum dependence of path length for particles with momenta near $p_\circ$. When rf is present, the energy E_{s} of a particle that arrives at location of the rf when the phase is $\varphi_{\mathrm{s}} = h\vartheta_{\mathrm{s}}$ such that $\Delta E_{\mathrm{s}} = \Delta E_\circ$ is called a synchronous particle, and φ_{s} is the synchronous phase. E_{s} is usually equal to $E_\circ$ so that the synchronous orbit and the central orbit are the same. Because this is typical, data for an rf system requires a harmonic number but not a frequency, because the rf frequency defaults to h times the the circulation frequency of a particle with momentum $p_\circ$. The program can calculate a ϑ value such that the particle having that phase will remain on the central orbit. When there is a single rf system, ϑ_{s} is the familiar synchronous phase below transition except for the h factor: $\vartheta_{\mathrm{s}} = \varphi_{\mathrm{s}}/h$. The synchronous phase is chosen as the origin for ϑ; thus, ϑ_{s} is the amount the waveform must be shifted so that the particle at $\vartheta = 0$ gets the synchronous energy increment. When there are several systems active, each may have its independent phase and harmonic number. However, if there is sufficient total amplitude, there is a phase by which the composite waveform can be shifted to give the synchronous energy increment at a phase stable point. This shift is still called ϑ_{s}, but it may not be the phase of any of the individual rf systems at the time the synchronous particle passes. Incidentally, the code generally proceeds as though all of the rf gaps are at one location. Usually the synchrotron tune is low enough, say

---

[1] Instructive examples of the capabilities of ESME can be found in reports relating to TeV I. Some of these will not be cited directly in the text but are included in the references. See for example refs. [2], [3], and [4].

$\nu_s < 0.005$, that this will be a good approximation. However, if the cavities are in fact distributed and the synchrotron tune high, one can get more realistic modeling by taking multiple energy kicks per turn; ESME allows a turn to be divided into any number of *identical* segments.

Because the quantities ϑ_i have the range $\pm\pi$, it is natural to identify them with the azimuth of the macroparticles at the synchronous time, *i. e.*, the time when the synchronous particle is at the rf location. However, because ϑ is actually rf phase, the conversion to azimuth depends on particle momentum. For particles close to the synchronous energy, the difference is qualitatively negligible, and useful intuition can be derived from picturing the distribution in ϑ as being distributed around the ring in the same way. Note, however, that in this picture smaller ϑ particles are leading particles; beam circulates in the ring in the $-\vartheta$ direction. Although it is not strictly correct, ϑ will frequently be referred to as the particle azimuth in the following. The conversion from the phase variable to the true azimuth $\Theta_{i,n}$ of the particle is

$$\Theta_{i,n} = \frac{\tau_{s,n}}{\tau_{i,n}}\vartheta_{i,n} \approx \left(1 - \eta\frac{\Delta p}{p}\right)\vartheta_{i,n} \quad .$$

For most applications to a high energy synchrotron the η-term is small compared to one. There may be some case for which the distinction between ϑ and Θ will be important.

The treatment starts from the specification of a reference orbit of average radius R_{eq} on which the mean normal magnetic field $<B_y>$ is known. A particle which would follow the reference orbit has the reference momentum $p_\circ = 2.997925{\cdot}10^2 <B_y> R_{eq}$ with $p_\circ$ in MeV/c, R_{eq} in m, and $<B_y>$ in Tesla. The angular frequency of beam circulation on this orbit is $\Omega_\circ = \beta_\circ c/R_{eq}$ where $\beta_\circ$ is the Lorentz β. The variation of the guide field away from the reference orbit is completely characterized for the purposes of describing longitudinal motion by the momentum dependence of the path length for fixed guide field. At a minimum one need only specify $\gamma_T = $ const., a choice adequate in many instances. In many problems it is also correct to identify the radius of the synchronous trajectory R_s with the reference radius R_{eq}. However, in applications like stacking or displacement acceleration where the synchronous trajectory may be offset radially, or even out of the beampipe entirely, it is necessary to take explicit account of the difference between them. ESME calculates all motion relative to a hypothetical synchronous particle. The synchronous frequency is calculated from momentum and radius. The momentum dependence of path length at the synchronous radius is determined from that given for the reference orbit.

It was mentioned before that the the synchronous energy E_s need not be equal to the reference energy $E_\circ$ on the central orbit. This is a minor generalization of what has been described. The mapping period is still the synchronous period. The path length dependence on momentum must be transformed from the reference momentum as an origin to the synchronous momentum. The synchronous energy can be separated from the reference energy by requesting a fixed energy offset (see 2.2.1). The rf data can include momentum offsets for individual rf systems, frequency programs, and phase programs which will move E_s away from $E_\circ$ (see 2.2.2).

The initial particle distributions can have any central energy. Naturally if a distribution is not centered on E_s, it will appear in different parts of the phase plane from turn to turn. Because the concept of a synchronous particle is fundamental to the program design, such a particle is usually tracked as a check on the program and the input data. A difference between the coordinates of this particle (EREF, THREF) and the presumed synchronous point (ES, 0) is an indicator of non-adiabatic change of parameters. Such difference could of course be intentional, but usually it is a sign that there is an error in the input data. It is also possible to initialize (EREF, THREF) at an arbitrary value for monitoring other aspects of the calculation.

1.2 Difference Equations

The basis of the program is the pair of single particle difference equations

$$
\begin{aligned}
\vartheta_{i,n} &= \left[\frac{\tau_{s,n-1}}{\tau_{s,n}}\vartheta_{i,n-1} + 2\pi\left(\frac{\tau_{i,n}}{\tau_{s,n}} - 1\right)\right]_{mod(\pi)} \\
E_{i,n} &= E_{i,n-1} + eV(\varphi_{s,n} + h\vartheta_{i,n}) - eV(\varphi_{s,n})
\end{aligned}
$$

giving the change in the azimuth and energy of particle i during the n-th turn of the synchronous particle. The n-th energy increment comes at the end of the n-th turn. The relation between the synchronous beam circulation period $\tau_{s,n}$ and that of the i-th particle $\tau_{i,n}$ is treated exactly. Thus, the kinematic nonlinearity is treated exactly; this feature can be very important if the synchronous energy is close to the transition energy. The lattice nonlinearity is expressed by expanding the trajectory length in powers of $\Delta p_i/p_\circ = (p_i - p_\circ)/p_\circ$ where p_i is the particle momentum and $p_\circ$ is the momentum on the reference trajectory. The rf potential is the sum of one or more sinusoidal terms so that the dynamic nonlinearity of a simple waveform is treated exactly and other waveforms are treated in a fourier expansion of ten or fewer terms.[2] The equation used in the program is generalized

---

[2] A little-used option mentioned in ref. [5] to provide a perfectly linear sawtooth waveform has been dropped because of severe conflict with the mechanism by which the more recent versions find the synchronous phase.

somewhat to permit multiple evenly spaced cavities per turn as an option and to allow the drift between cavities to be subdivided for more frequent application of the space charge kicks if the dynamics require it.

There is a problem of numerical instability above transition when the synchronous phase is defined in the customary way. The source of the difficulty is the truncation error in the argument to the sine function(s) in the map. With 64 bit real numbers the instability becomes apparent somewhere in the vicinity of 10^6 to 10^7 iterations in test cases on a stationary bucket. Going to quad precision postpones the problem for another order of magnitude or more but does not eliminate it entirely. There is a good way to monitor a calculation for signinicant effects of the instability: make a HISTORY plot of average energy for the distribution (EBAR) *vs.* average phase (THBAR) and look for the development of a systematic spiralling growth of the amplitude of the center of charge oscillation. In the tests run, the effect shows up at the 10^{-16} level after thousands of iterations and is often still below significance beyond 10^6 iterations. The instability has always been present but has not been a practical concern for typical uses of the code. However, it came to light in some experiments on the onset of Robinson instability where the parameters were intentionally close to threshold. There seem to be four ways to deal with it:

1. Ignore it; it never bothered anybody before.

2. Monitor it by a HISTORY plot in cases where it could be manifest.

3. Use quad precision in sine and in accumulating its arguement.

4. Modify the map so that the stable fixed points are limited to the range $-\pi/2 \leq \varphi_s \leq \pi/2$.

Because the sine calls are burried within the very tightest inner loop, as many as ten calls per particle per turn, the question of computing time is very important in choosing the response to the instability problem, and the choice between options 2 and 4 is arguable. Option 1 can be accepted where fewer than 10^4 iterations are used, but it is not very satisfying for a code that is intended to serve the widest possible range of applications within a uniform framework. For option 3 one pays an unacceptable price in computer time, rougly a factor of 30 in a couple of trials. The great virtue of option 4 is that it fundamentally fixes the problem and lifts the burden of vigilence from the user. The price is a somewhat disconcerting definition of the synchronous phase above transition. By changing a sign in the map above transition, the particles can be made to follow the correct trajectories with the stable point on the positive slope of the waveform. Thus in this case there is no replacement of the stable phase below transition by its supplement above transition. There is no doubt that this option is superior from the standpoint of numerical analysis. Because no single option seems fully satisfactory, the source includes a preprocessor choice (*cf.* section 4.2) to compile either with the map given above or with the intrinsicly stable form. In compiled distributions, the unmodified map is used.

In the section on parameters for the tracking routine there is mention of a parameter for choosing alternative versions of the difference equations. There are two choices, one which calculates $\Delta\vartheta$ as indicated above and one that equates it to the familiar approximation $-2\pi\eta\Delta p/p$. Substantial time can be saved by using the simple difference equation if it is sufficient.

1.3 Multiparticle Calculations

The particle distribution at each iteration is mapped from the previous by the difference equations. The program provides three general types of optional calculation on the properties of the distribution as a whole. The most common calculations are those which quantify properties of the distribution so that one can plot them as functions of time. Examples of such properties are first and second moments, emittance, fourier spectrum of beam current, *etc.* Another type of calculation involves calculating feedback contributions to rf system parameters. ESME provides for phase feedback and feedback to rf amplitude. The third general class of collective calculation is the evaluation of beam induced voltages from space charge and longitudinal coupling impedance. Included in this category are provisions for feeding back or forward to correct beam loading phase shifts of the rf cavities. The space charge calculation uses a geometric factor depending on average beam radius and average beampipe radius. This factor is rolled off at very high frequencies. The longitudinal impedance can be characterized by an arbitrary table of real and imaginary part *vs.* frequency and/or a table of resonances defined by resonant frequency, Q, and real part at resonance. The frequency domain treatment of bunch self force and wall impedance is described in ref. [7]. Several interesting applications were made to the Fermilab Booster, which are described in refs. [8, 9, 10, 11].

When a resonance is represented by an impedance, the transient part of the excitation is neglected. However, this treatment does not realistically represent the driving terms for certain types of collective bunch motion. Therefore, response of a high-Q resonator may also be modeled in the time domain (see ref. [12]). Several resonances can be treated simultaneously with some (or all) treated in time domain and some (or all) treated in frequency domain. Reference [13] describes the use of this facility in modeling the coupled bunch instability. The code also provides a more general time domain calculation of the effect of wake fields. The calculation is based on a basis function giving the voltage response [MV] to a unit charge [C] distributed in a triangular pulse.[14] The basis function is calculated at an interval small enough to give good linear interpolation on the charge distribution and carried out to times for which the total wake field will be significant but in no case for more than one

turn. This option may be used along with both the frequency domain calculation and the time domain calculation for simple resonances. The bunch self-force, *i. e.* the perfectly conducting wall term, can be calculated using either the fft machinery used in the impedance calculation or in the time domain with a locally determined numerical derivative of the azimuthal charge distribution.

Calculations of collective effects often require many particles to avoid spurious instabilities driven by numerical noise. If the aim is to model only low frequency effects like bucket distortion or low frequency parasitic modes, it may be acceptable to smooth the charge distribution using either local smoothing or the global (Berstein polynomial) options provided. However, an arguably more general and rigorous approach, which does not arbitrarily sacrifice high frequency information, is to scale the map[1] with the effects that both fewer iterations and fewer macroparticles are required. The code is supposed to scale all impedances automatically, but a *soupçon* of scepticism is salutory.

1.4   Flow Lines in Phase Space

Although the emphasis during the development of ESME has been on the evolution of distributions, the plotting of flow lines is a complementary approach to visualizing phase space motion which can either replace or enhance scatter plots of the distribution. There are several variants on the contour plotting approach.

1. At selected intervals through the acceleration cycle one can produce a flow line map. If a distribution is tracked, it may be plotted along with the flow lines or its plotting can be suppressed. If no distribution is tracked between the times at which the flow line displays are generated, the flow lines can not reflect the action of a collective potential.

2. An initial set of flow lines can be generated and then tracked as a distribution itself, either with or without an accompanying finite emittance distribution. If a collective potential is calculated with such a composite distribution, the flow line component is ignored, but the evolution of the flow line component will include the effect of the collective potential.

3. A fine grain visualization of the result of non-adiabatic motion can be obtained by tracking a flow line type of distribution and plotting it along with flow lines representing the final system parameter values. The regions of phase space where non-adiabatic motion has been significant will show up as areas where the tracked and the freshly generated contours differ.

4. A specialized variant of the tracking of a contour-like distribution is available as an option to the command which sets up the initial phase space distribution. A distribution of particles just inside and just outside the bucket is available. It can be helpful, for example, in understanding the effects of small perturbations on particle loss.

5. Flow lines can be generated from several arbitrary starting points.

6. Flow lines may be either at fixed parameters, representing the hamiltonian flow at a particular time or with parameters that vary during the time the particle generates the trajectory to illustrate, for example, the effects of a parametric resonance.

The difference equations are used to generate lines of flow in phase space. If the flow lines are to reflect the influence of the collective potential derived from the charge distribution, there is an intrinsic contradiction when the longitudinal coupling impedance $Z_{||}$ has a real part because parameters change with time as the trajectory generating particle delivers energy to the impedance. Such a spiralling flow line does not represent what is happening to particles in the distribution because, in the usual case, the synchronous phase is adjusted slightly to keep the the bunch on the reference orbit. Therefore, when a single particle trajectory is calculated for a fixed time, the collective potential is reduced by its average value so that bounded trajectories will close. In the absence of a real component of impedance, the flow lines at fixed time are exactly lines of Hamiltonian flow.

Chapter 2

Program Organization and Data Requirements

Data for ESME are generally acquired by NAMELIST reads dispersed among subroutines which segregate different program functions as much as practical. The subroutine reading a particular class of data will be called only if there is in the input stream a single-character command requesting the related program function. The data relating to different program functions or accelerator subsystems are also stored in different FORTRAN COMMON blocks. These blocks are initialized with values or switches to allow the code to proceed on the basis of a few input data. The initial values, or those read in over them, are retained unaltered[1] so that a command may be repeated without reentering data that remain *apropos*. There are default values provided for many parameters to increase the likelihood of completing execution even if the input data fail to fully define the problem.

The following subsection calls attention to requirements on command ordering arising from data dependencies. Next is a subsection listing all of the ESME command characters with a short description of their functions. They are listed in the order that would be likely to appear in an input data set, although some are not affected by order of appearance. The subsection following the listing of the command codes contains NAMELIST's associated with the commands and a brief description of the function of each input datum. The same tabular information is collected in more compact form in Appendix C where the commands are arranged in alphabetical order for convenient reference.

2.1 Command Ordering

The order in which the commands may appear in the data will generally be logically apparent. For example the **R** command, which initiates the input of the basic lattice parameters and energy scale, usually will appear before any other command. Several commands require this information to perform their own functions. The **A** command, which brings in the rf parameters, is generally the second command to appear. The command **P**, which establishes the initial phase space distribution, generally needs to be preceded by both. If memory allocation differing from the default setting is used, the **Y** command must be called before the initial distribution is stored. The command **B**, which sets up the machinery for longitudinal impedance calculations, needs the initial distribution to get the initial beam current distribution. Although by detailed knowledge of the internal organization of the program or by exploiting some specialized facilities it is possible to create correct data sets that do not observe a typical command ordering, it is generally safer to adhere to the order **Y,R**, **A**, **P**, **B** of first appearance for these commands. The **C** command, which generates phase space flow lines, must follow **R** and **A**; if the flow lines are to take account of the collective potential produced by a bunch, it must follow **P** and **B** also. Other commands can usually be ordered arbitrarily.

2.2 Commands

The principal functions of the program are activated by single-character commands appearing in the first position of a data record. Starting in the fifth position is an optional character field which serves to annotate the dataset itself and appears in the program output as a useful indication of the motivation for each step. The commands initiate a problem data read and/or an action upon available data. If data are requested by a subroutine invoked by the command, data in NAMELIST form will follow in the next record(s). Character strings like the plot title or the names of auxiliary files are read as string variables. Exceptions are the file names for the save/restore commands (**S** and **G**) which are read with an A format and possibly data requested by user-written SHAZAM routines (**0** – **9** commands).

The data requested or process initiated by a command is related to a separable subsystem of the accelerator or a distinct phase of the calculation. The course of program execution is governed by the order of the commands in the data stream; that is, the program is what is sometimes called "data-driven". Since the data are stored in functionally segregated common blocks, any subroutine that declares the appropriate common has access to any input datum.

---

[1] There are exceptions where a parameter is reset after it has taken effect to avoid undesirable repetitions.

Commands are listed below in four groups. The first group consists of the commands that will appear in almost every problem data set. They are listed in the order in which they are generally used. Interesting calculations may be made using these commands only. The second group contains commands for additional output options. Those in the third group invoke special calculations, including, if desired, ones written by the user. The fourth group consists of a save/restore pair for the entire state of the calculation and a dynamic memory allocation command. Commands with strong order dependence are underlined. In the description of a command the first letter of a keyword is indicated in bold face to call attention to an association between the command character and its function. Because of evolution of the code over twenty years some of the mnemonic connections are a bit weak, but it is useful to resurrect old files occasionally. Thus, continuity is the more important consideration.

List of Commands

R read in the lattice (**R**ing) parameters, magnetic field ramp, problem energies, *etc*. according to NAMELIST /RING/.

A read in the radio frequency (**A**cceleration) parameters according to NAMELIST /RF/.

P **P**opulate the phase space with the initial distribution described by parameters read according to NAMELIST /POPUL8/.

O Select graphical **O**utput options using parameters in NAMELIST /GRAPH/.

T **T**rack distribution according to parameters read in NAMELIST /CYCLE/.

Q **Q**uit the program.

....

D **D**isplay graphical output.

W **W**rite comment into printed output.

H Select quantities to be plotted from **H**istory records according to NAMELIST /HISTRY/.

M Save azimuthal histograms of distribution for composition of **M**ountain-range plot.

N Plot mountain-range data.

....

L Set **L**ow-level parameters (to control feedback, transition phase change, *etc*.) in NAMELIST /LLRF/.

B Setup collective potential calculation; read in **B**eam parameters in NAMELIST /SCHG/.

F Setup the fast **F**ourier transform of the theta distribution, control parameters in NAMELIST /FFT/.

C Form flow line **C**ontours at time intervals or as initial distribution, control parameters in NAMELIST /FLOW/.

K **K**ill all or selected parts of the phase space distribution according to control parameters in the NAMELIST /KUTS/.

0-9 Enter subroutine SHAZAM at SHAZAM, SHAZAM1, SHAZAM2, … to manipulate any quantity in COMMON storage in the program.

....

Y Allocate memor**Y** to certain arrays according to input data in NAMELIST /MEMORY/.

S **S**ave tracking and control data.

G **G**et tracking and control data from a previous run.

2.2.1 <u>R</u> Command - Lattice parameters

	Default		
Variable	Value	Unit	Description
REQ	None	m	The reference radius for the central orbit
GAMTSQ	None	-	Square of transition γ (< 0 accepted)
ALPHA1	0.0	-	Coefficient of $(\Delta p/p)^2$ in series expansion for length difference between particle trajectory and reference orbit
ALPHA2	0.0	-	Coefficient of $(\Delta p/p)^3$ in series for path length difference
ALPHA3	0.0	-	Coefficient of $(\Delta p/p)^4$ in series for path length difference
W0I	None	MeV	Kinetic energy on the central orbit at T = TI
W0F	0.0	MeV	Kinetic energy on the central orbit at T = TF (*see text*)
P0I	None	MeV/c	Momentum on the central orbit at T = TI
P0F	0.0	MeV/c	Momentum on the central orbit at T = TF (*see text*)
TI	0.0	s	Start time of magnetic field change
TF	0.0	s	End time of magnetic field change
TSTART	0.0	s	Time at which tracking begins
FRAC	1.	-	Determines azimuthal periodicity, calculation restricted to $-180°/\text{FRAC} \le \vartheta \le 180°/\text{FRAC}$
NCAV	1	-	Number of rf locations (default usually adequate)
PIPRAD	1.0	m	Radius of beam pipe for particle loss
EBDRY	F	-	Switch to set absorbing beam pipe walls at REQ $\pm$ PIPRAD
DES	0.0	MeV	Constant energy offset of synchronous orbit relative to reference orbit
KURVEB	1	-	Magnetic field ramp from W0I to W0F or P0I to P0F: 1 – Linear 2 – Increasing parabolic 3 – Biased sinusoidal (See text for definitions CRA, CRB, CRC, TR3.) 4 – Arbitrary ramp table read from file given by FILRMP 5 – Parabolic with final slope W0FDOT 6 – Parabolic with final slope P0FDOT 7 – Cubic with zero initial slope and curvature
			8 – Decreasing parabolic
CRA	0.0	-	Time shift of first harmonic of KURVEB=3 ramp
CRB	0.0	-	Relative amplitude of second harmonic in KURVEB=3
CRC	0.0	-	Time shift of second harmonic for KURVEB=3
TR3	0.0	s	Half-period of fundamental for KURVEB=3; TR3=0. $\Rightarrow$ half-period = TF-TI
W0IDOT	0.0	MeV/s	Slope of parabolic ramp at TI (KURVEB=5)
W0FDOT	0.0	MeV/s	Slope of parabolic ramp at TF
P0IDOT	0.0	MeV/c/s	Slope of parabolic ramp at TI (KURVEB=6)
P0FDOT	0.0	MeV/c/s	Slope of parabolic ramp at TF
JNRAMP	F	-	Establishes starting point of ramp as point at which program finds itself–for smoothly piecing ramp segments together
FILRMP	'DUMMY'	-	Name for file containing ramp table (KURVEB=4) (see 3.2.3)
GMAJMP	F	-	Set γ_T-jump on (*See text for compatibility.*)
KINDG	1	-	Type of γ_T variation: [a] 1 – Linear ($\gamma_\text{T} = \text{GAMPAR}(1) + \text{GAMPAR}(2) * \text{T}$) 2 – Decreasing exponential ($\gamma_\text{T} = \text{GAMPAR}(1) + \text{GAMPAR}(3) * (1 - e^{-\text{T}/\text{GAMPAR}(2)})$)
GAMPAR(1:3)	0.0	-	Coefficients for γ_T variation (real γ_T only)
CHGNO	1.0	-	Charge of beam particle in units $|e|$
EM0CSQ	938.27231	MeV	Rest energy of beam particle

<div style="text-align:center">R Command, Namelist /RING/</div>

[a]T $= 0$ corresponding to time at which **R** command is invoked with GMAJMP = .TRUE.

The members of NAMELIST /RING/ are stored in COMMON /RINGP/ by SUBROUTINE RINGPAR, which also derives

quantities depending only on lattice parameters. It is possible to distinguish the reference trajectory, of energy E_0, from the synchronous trajectory, of energy E_s, by specifying an offset DES from the reference energy. An example of a simple **R** command is

$$\boxed{\text{\$RING REQ=1000. GAMTSQ=351.56 W0I=150000. FRAC=159. \$END}}$$

in which the lattice is characterized simply by a radius, γ_T value, and a field which corresponds to a reference energy of 150 GeV. FRAC is useful for restricting the range of consideration of ESME to a suitable period of the ring. In this case, since W0F is not specified, the default, W0F = 0.0, indicates that the guide field does not change. In addition, the start time for this run is taken to be zero. Notice that it is possible to treat a lattice with imaginary γ_T simply by entering a negative value for GAMTSQ.

For time $\leq$ TI the energy of the reference orbit is taken to be W0I, while for time $\geq$ TF the energy of the reference orbit is W0F. Either or both quantities W0I,W0F (kinetic energy) may optionally be replaced by P0I,P0F (momentum). Note that PIPRAD has no effect unless EBDRY = T.

Resonant fast cycling synchrotrons like the Fermilab booster have a practically sinusoidal ramp

$$B(t) = B_i + (B_f - B_i) \left[1 - cos\left(\pi \frac{t - t_i}{t_f - t_i} \right) \right] / 2 \quad .$$

This is the curve one gets for KURVEB=3 if the four constants CRA, CRB, CRC, TR3 in the **R** command are all left at the default value of zero. Take a time variable $x = (t - t_i)/(T - t_i)$ in which the half period T is TR3 for TR3 $\neq 0$ and $t_f - t_i$ otherwise. The full expression for KURVEB=3 is

$$B(t) = B_i + (B_f - B_i)\frac{-\cos \pi(x + a) - b\cos 2\pi(x + c) + \cos \pi a + b\cos 2\pi c}{2\cos \pi a} \quad .$$

The fraction is 0 for $x = 0$ and 1 for $x = 1$. The data item CRB is the constant b in the above expression; it allows the addition of second harmonic for such purposes as reducing rf power requirements for the cycle. The constant a (CRA) permits starting the ramp with a slope as one might want to do simulating injection starting before $\dot{B} = 0$. CRC could be used to move the second harmonic along with the first or, perhaps, to turn the cos into a sin. The half-period value TR3, if specified, allows the use of sinusoidal ramps with a time scale independant of the times TI and TF bounding the period in which the curve is used. Notice that if the half-period is not $t_f - t_i$, B_f is not the magnetic field at the end of the ramp; it is rather the maximum value, the value the field will have at t=T.

Perhaps the KURVEB=7 option seems a little obscure. This curve segment has been used to represent what is called the voltage parabola for the Fermilab main ring power supplies, which did not like to turn on with a finite voltage slope; presumably other high current supplies have similar properties.

If the γ_T-jump option has been invoked, then the γ_T is varied until the **R** command is issued with GMAJMP = F; it applies only to real γ_T. It is a special purpose option (euphemism for kluge) which is incompatible with all other uses of the **R** command. If it is necessary to change other lattice parameters at the time when a γ_T-jump is also to be initialized, the ordinary lattice parameters should be entered in a first **R** command with GMAJMP = F; the jump should then be set up in a second command with GMAJMP = T.

Occasionally it is useful to model the steps immediately preceding and following a transfer between accelerators in a single ESME run. If all of the parameters for the second machine are included in new **R**, **A**, **B**, *etc.*, the simulation will proceed from machine to machine. The distribution will be scaled in ϑ to account for change in the reference orbit radius R_{eq}.

2.2.2 <u>A</u> Command - RF parameters

	Default		
Variable	Value	Unit	Description
NRF	1	-	Number of active RF sources
H(1:10)	1	-	Harmonic numbers of sources (integers)
HW(1:10)	1	-	Voltage sources will be expressed over the ϑ range $-180°/\text{HW} \leq \theta \leq 180°/\text{HW}$ to produce isolated buckets or barriers
ISYNC	0	-	Indicates synchronism condition to be imposed on RF: 0 – None, voltages and phases remain as programmed 1 – Phase of RF waveform shifted to synchronous stable point 2 – RF amplitude scaled to give synchronous energy with no change of phase 3 – Source 2 Landau cavity to source 1, synchronism assured only for sources 1 and 2
VI(1:10)	0.0	MV	Voltage of source I at time TVBEG(I)
VF(1:10)	0.0	MV	Voltage of source I at time TVEND(I)
TVBEG(1:10)	0.0	s	Time corresponding to beginning of RF voltage change
TVEND(1:10)	0.0	s	Time corresponding to end of RF voltage change
KURVE(1:10)	0	-	Specifies type of RF voltage variation between times TVBEG and TVEND: 0 – None, voltage maintained at VI(I) 1 – Linear 2 – Iso-adiabatic 3 – Sigmoid 4 – Fit and cubic spline interpolation of voltage table[a]
VKON	T	-	Indicates whether programmed voltage curves are to be active
PSII(1:10)	0	deg	Phase of source I at time TPBEG(I)
PSIF(1:10)	0	deg	Phase of source I at time TPEND(I)
TPBEG(1:10)	0.0	s	Time corresponding to beginning of RF phase change
TPEND(1:10)	0.0	s	Time corresponding to end of RF phase change
KURVP(1:10)	0	-	Specifies type of RF phase variation between times TPBEG and TPEND: 0 - None, phase maintained at PSII(I) 1 - Linear 2 - Quadratic 4 - Fit and cubic spline interpolation of phase table[a]
PHKON	F	-	Indicates whether or not phase curves are to be active
FRI(1:10)	0.0	MHz	Frequency of source I at time TFBEG(I)
FRF(1:10)	0.0	MHz	Frequency of source I at time TFEND(I)
TFBEG(1:10)	0.0	s	Time corresponding to beginning of frequency change
TFEND(1:10)	0.0	s	Time corresponding to end of frequency change
KURVF(1:10)	0	-	Specifies type of frequency variation between times TFBEG and TFEND: 0 - None, frequency maintained at FRI(I) 1 - Linear 2 - Quadratic 4 - Fit and cubic spline interpolation of frequency table[a]
FRKON	F	-	Indicates whether frequency curves are to be active
FILCRV	'DUMMY'	-	Full path for rf curves file
CNTINU(1:10)	F	-	Sets the starting voltage, phase and/or frequency for the corresponding sources to the current values — for smoothly piecing curve segments together
VMATCH(1:10)	F	-	For source I, $\text{VMATCH}(I) = T$ results in the VI(I) being set so that source I is matched to the current distribution emittance [b]

A Command, Namelist /RF/

---

[a] Fit to values read from file. See Section 3.2.2.

[b] Which means, in this instance, that a P command, or its equivalent, should precede the A command.

	Default		
Variable	Value	Unit	Description
HOLDBH	F	-	Switch to calculate the voltage of *principal source*[a] to hold the current bucket height. (see also HDECR)
HDECR	1.0	-	Factor by which bucket height for *principal source*[a] is to be adjusted on successive iterations if HOLDBH = T
HOLDBA	F	-	Switch to calculate the voltage of *principal source*[a] to hold the current bucket area. (see also SDECR)
SDECR	1.0	-	Factor by which bucket area for *principal source*[a] is to be adjusted on successive iterations if HOLDBA = T
PHISLIM	.95	-	Voltage may not be reduced such that $\sin \phi_s >$ PHISLIM using options HOLDBH and HOLDBA
PHSLIP	F	-	Switch indicating that the phase of at least one source is to be varied to correspond to a momentum offset from the synchronous value (see DELTRF)
DELTRF(1:10)	0.0	-	Momentum offset $(\Delta p/p)$ at which source I is to be operated

<div style="text-align:center">A Command, Namelist /RF/, continued</div>

[a]The constant bucket area or bucket height is calculated from the rf source which gives the greatest bucket height.

The members of NAMELIST /RF/ are read in SUBROUTINE RFPROG, and stored in COMMON /RFP/. Up to 10 independent voltage sources may be specified. An example of a simple **A** command is

$RF H=1113 VI=.100 PSII=-55.0 $END

Here, one voltage is specified by a minimum set of parameters. Note that H(I) is an integer (as is HW(I)). Since RF manipulations are at the heart of ESME, this command can become rather lengthy and involved; the example given above is exceptionally brief. As in the case of the magnetic field ramp, specified in the **R** command, the relevant quantity is maintained at the initial value at times prior to the start of a programmed variation, while at times after the end of a programmed curve, the quantity is maintained at the final value. This makes it easier to specify multiple curves in unequal time steps. Particular attention is called to the CNTINU parameters which help join segments smoothly by setting initial values to current values.

The values of the programmed phase curves are to be distinguished from the phase of a given harmonic at the synchronous particle, though they may be the same. For example, a user could represent any periodic waveform (in a Fourier expansion of up to 10 terms) simply by specifying the correct relative phases and amplitudes of the voltages. The waveform could be modified over time by specifying the variations of the voltages, phases, and/or frequencies.[2] Generally, the user will specify that the program search for a synchronous point on the resultant waveform. Option ISYNC = 1 will result in the program searching for a stable value of the phase, stored internally in the program as PHIS, a phase at harmonic number one, and output as degrees of shift of the sum voltage waveform. Option ISYNC = 2 will cause the resultant waveform to be scaled to give only the correct magnitude of voltage at the synchronous particle, not necessarily at a stable slope, since the phases of the voltage sources remain at their programmed values. The Landau cavity option, ISYNC = 3, will vary the phases of voltage sources 1 and 2, and the magnitude of voltage source 2, so that the first and second derivatives of the voltage vanish at the synchronous particle. The phases and magnitudes of other voltages are not altered, though they will be included in the iteration of the difference equation if NRF > 2. As noted in the table, the options activated by HOLDBH and HOLDBA are calculated from the bucket height or bucket area respectively of the system which gives the greatest bucket height, considered to be the principal system. The voltages for all systems are scaled by the ratio found for the voltage of the principal system. These options provide an extremely convenient way to generate automatically a reasonable voltage program for any ramp curve. They require ISYNC=1.

---

[2]In cases in which a frequency and a phase variation are specified for the same source, the phase variation takes precedence.

2.2.3 <u>P</u> Command - Initial distribution parameters

	Default		
Variable	Value	Unit	Description
KIND	1	-	Chooses the type of distribution to be generated: 1–Rectangular outline, NTH by NE points, limited by THMIN, THMAX, REMIN, REMAX 2–Uniform rectangular grid NTH by NE, limits as in KIND = 1 3–Random uniform distribution of NPOINT points within rectangular limits as in KIND = 1 4–Random uniform in θ, limits THMIN, THMAX; Gaussian in E, REMIN, REMAX = $\pm 2\sigma$, NPOINT points 5–Gaussian in θ, THMIN,THMAX = $\pm 2\sigma$; random uniform in E, limits REMIN, REMAX, NPOINT points 6–Rectangular grid, regular in θ, Gaussian in E, NTH by NE points .... The remaining distribution types, except for 11, 12, 15, and 20 are *matched*, the distribution limited by a contour of SBNCH eVs. .... 7–Bunch outline of NPOINT particles 8–Regular grid of approximately NTH by NE particles 9–Random uniform bunch of NPOINT particles within contour 10–Bi-Gaussian distribution of NPOINT particles, 95% within contour 11–NPOINT uniformly spaced particles on flow lines just above and below bucket boundary 12–NPOINT particles, random uniform in E, limits REMIN,REMAX, parabolic in θ, limits THMIN,THMAX 13–Parabolic distribution of NPOINT particles 14–Elliptical distribution of NPOINT particles 15–NPOINT particles, random uniform in θ, limits THMIN,THMAX; parabolic in E, limits at REMIN,REMAX 16–NPOINT particles within matched contour, low-noise uniform quasi-random 17–NPOINT particles within matched contour, low-noise gaussian quasi-random 18–NPOINT particles in matched contour, low-noise uniform pseudo-random (Sobel sequence) 19–NPOINT particles in matched contour, low-noise gaussian pseudo-random (Sobel sequence) 20–NPOINT particles, pseudo-random within limits REMIN,REMAX, THMIN,THMAX
NPOINT	1	-	Number of particles generated for all distributions except KIND = $1, 2, 6, 8$, in which NTH and NE are used
NTH	2	-	Number of grid points in θ direction
NE	2	-	Number of grid points in E direction
THMIN	-90.0	deg	Lower θ limit on rectangular distributions
THMAX	90.0	deg	Upper θ limit on rectangular distributions
REMIN	None	MeV	Lower energy limit on rectangular distributions; relative to the synchronous energy, ES
REMAX	None	MeV	Upper energy limit on rectangular distributions

Variable	Default Value	Default Unit	Description
SBNCH	0.1	eVs	Area within limiting contour
IPOP	1	-	Specifies RF source for generating bunch limiting contour: 0–All active (NRF) sources I–Source I ($1 \leq I \leq$ NRF)
THOFF	0.0	deg	Amount to displace distribution generated in current call to POPUL8 in θ direction
EOFF	0.0	MeV	Amount to displace currently generated distribution in E direction
THTRAN	0.0	deg	Amount to displace all particles (generated in this and previous calls to POPUL8) in θ direction
ETRAN	0.0	MeV	Amount to displace all particles in E direction
WINJ	0.0	MeV	Kinetic energy of injection; if both WINJ and PINJ are at the 0.0 default value, injection energy is $E_\circ$ of the central orbit
PINJ	0.0	MeV/c	Injection momentum, alternative to WINJ above (*q. v.*)
PARTION	F	-	Partition distribution into separate classes;[a] each separate use of the **P** command with PARTION = T introduces a new partition
RENORM	F	-	Calculate ANORM for matched bunch so that EPSILON $\equiv$ SBNCH. RENORM is reset to .FALSE. at each use.

[a]Different classes of particles may be plotted with distinct symbols.

The members of NAMELIST /POPL8/ are read in SUBROUTINE POPUL8, and stored in COMMON /POPLATE/. To populate a bi-Gaussian distribution of 100 particles, with an emittance (95%) matched to RF source 1 of .02 eV-s, the **P** command would be

$POPL8 KIND=10 NPOINT=100 IPOP=1 SBNCH=.02 $END

For multi-bunch simulations, in which several distributions are needed with the same parameters but in different positions, it is sufficient to simply re-issue the **P** command with only the desired position offset. For example

$POPL8 THTRAN=45.0 ETRAN=0.0 $END

Since NAMELIST members which do not appear in the input remain unchanged, one can limit the input by giving only the parameters with changed values. In particular, THTRAN and ETRAN make it unnecessary to explicitly declare the position of each group of particles.

Setting RENORM .TRUE. will result in the inclusion of a factor in the emittance which scales the rms emittance of the intial bunch to its SBNCH value. So long as the bunch form remains similar, the emittance so calculated gives an approximate value for the total bunch area. Generally it seems safer to use the unscaled rms emittance.

2.2.4 O Command - Graphical output options

Variable	Default Value	Unit	Description
	Default		
Variable	Value	Unit	Description
MPLOT	1000	-	Output every MPLOT iterations
IDEV	1	-	Virtual device number for graphical output
POSTP	F	-	Write all data in COMMON blocks to unit 18; do not call plotting routine.
TITL	exists	-	String of up to 50 characters for plot headings
PLTSW			Select plot options:
(1)	T	-	Draw phase space plot
(2)	T	-	Plot phase space points (different symbol for each class)
(3)	F	-	Interconnect points within each class
(4)	F	-	Draw lines at centroid and $\pm\sigma$
(5)	F	-	Draw voltage waveform
(6)	F	-	Set plot boundaries to turning points of contour
(7)	F	-	Suppress captions, axis labels, etc.
(8)	T	-	Plot θ histogram
(9)	F	-	Set θ histogram limits to turning points of contour
(10)	T	-	Plot E histogram
(11)	F	-	Set E histogram limits to turning points of contour
(12)	F	-	Plot fourier amplitudes
(13)	F	-	Include phases in plot of fourier spectrum
(15)	F	-	Plot the real and imaginary impedance from the input
(17)	T	-	Start bucket contour at unstable fixed point [a]
(18)	T	-	Start bucket contour above stable fixed point at $E = E_s + H_{bckt}$
(19)	F	-	Plot flow line points (different symbol for each class)
(20)	F	-	Interconnect flow line points within each class
(21)	F	-	Bunch perf. cond. wall voltage *vs.* ϑ
(22)	F	-	Include distribution in previous plot
(23)	F	-	Wake field (time domain) voltage *vs.* ϑ
(24)	F	-	Include distribution in previous plot
(25)	F	-	Frequency domain voltage *vs.* ϑ (inc. perf. cond. wall if FDSCON = T)
(26)	F	-	Include distribution in previous plot
(27)	F	-	Voltage from special time domain calculation for resonators em vs. ϑ
(28)	F	-	Include distribution in previous plot
(29)	F	-	Create don't-plot area in phase plane defined by four points[b]
(30)	F	-	Create don't-plot area in phase plane defined by matched contour[c]
(31)	F	-	PLTSW(31)=T defines don't-plot area *outside* closed figure
(32)	F	-	Net beam loading voltage *vs.* ϑ.
(33)	F	-	Plot current (net of beam, feed-forward, & feedback) driving cavity
NPJMP	1	-	In phase space plot, plot only every NPJMPth point
KLPLOT	0	-	Select classes in phase space plot and projections (see Sec. 2.2.3)
			0 — All classes plotted
			$1 \le$ KLPLOT $\le$ KLASSES — Plot class KLPLOT only
IRF	1	-	Selects voltage source for contour plotting:
			< 0 — No contour plotted
			0–All active (NRF) sources
			1-10 — Source IRF ($1 \le$ IRF $\le$ NRF)

O Command, Namelist /GRAPH/

[a] At least one of PLTSW(17) or PLTSW(18) must be true for ICONTUR=1 else program sets both .TRUE.
[b] see definitions of THEXCMI, THEXCPL,DEEXCMI,DEEXCPL
[c] see definition of SEXCL

	Default		
Variable	Value	Unit	Description
ICONTUR	1	-	Select the type of reference contour to plot on phase space plot
			0 — No contour
			1 — Bucket contour
			2 — Contour of initial bunch area SBNCH
			3 — Contour of the specified area REFAREA
			4 — Contour containing 95% of the particles
			5 — Flow lines chosen by LINES and STVAL
REFAREA	0.1	eVs	Area of reference contour for ICONTUR = 3
LINES	1	-	Number of flow lines for ICONTUR = 5
STVAL	0.,0.	[a]	Starting values for contours, LINES coordinate pairs in θ, E
THPMIN	0.0^b	deg	Lower θ limit for phase space plot
THPMAX	0.0^b	deg	Upper θ limit for phase space plot
DEPMIN	0.0^c	MeV	Lower E limit for phase space plot
DEPMAX	0.0^c	MeV	Upper E limit for phase space plot
IEREF	1	-	Determines energy origin for phase space:
			1 — E0, the reference energy (often = ES)
			2 — ES, the synchronous energy
			3 — EBAR, the average particle energy
			4 — EREF, the reference particle energy[d]
NBINTH	50	-	The number of bins for the θ histogram
THBMIN	0.0^e	deg	Lower limit for θ histogram
THBMAX	0.0^e	deg	Upper limit for θ histogram
NBINE	50	-	The number of bins for the E histogram
EBMIN	0.0^f	MeV	Lower limit for E histogram
EBMAX	0.0^f	MeV	Upper limit for E histogram
IFBMIN	1	-	Lower limit for FFT plot - If specified, specify IFBMAX also
IFBMAX	0^g	-	Upper limit for FFT plot
CVBMIN	0.0^h	deg	Lower θ limit for all collective voltage plots
CVBMAX	0.0^h	deg	Upper limit for all collective voltage plots
THEXCMI	0.	deg	Lower theta value defining rectangular don't plot area
THEXCPL	0.	deg	Upper theta value defining rectangular don't plot area
DEEXCMI	0.	MeV	Lower energy value defining rectangular don't plot area
DEEXCPL	0.	MeV	Upper energy value defining rectangular don't plot area
SEXCL	0.	eVs	Area of matched contour defining a don't-plot area
DTHCURV	0.0	deg	Amount by which contour will be moved in θ direction
DECURV	0.0	MeV	Amount by which contour will be moved in E direction

O Command, Namelist /GRAPH/, continued

[a]Starting values for contours have units degrees,MeV.

[b]THPMIN and THPMAX both 0.0 results in a plotting range $-180°/\text{FRAC} \leq \theta \leq 180°/\text{FRAC}$.

[c]DEPMIN and DEPMAX both 0.0 results in a plotting range approximately the range of particle energies

[d]A particle which ESME tracks from the origin (0,ES) as a reference.

[e]Limits of 0.0 for both THBMIN and THBMAX result in the plot range being the same as for the phase space plot.

[f]Limits of 0.0 for both EBMIN and EBMAX result in the plot range being the same as for the phase space plot.

[g]IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

[h]Limits of 0.0 for both CVBMIN and CVBMAX result in the range for the plot being $\pm 180°/\text{FRAC}$.

O Command, Namelist /GRAPH/, continued			
Variable	Default Value	Unit	Description
DELCON	.01	-	Determine bucket to precision DELCON*360° w/ RF
KNTLIM	500000	-	Number of iterations of difference equation which will be attempted to close contour
IMETA	0	-	Directs graphic output < 0 - Output written to a metafile 0 - Output to device selected by IDEV > 0 - Output to both metafile and selected IDEV
XCRNR	135.	0.001	Fraction of full width of plot frame between lefthand edge and left side of plot
YCRNR	69.0	0.001	Fraction of full height of plot frame between bottom edge and bottom of plot
XAXISL	750.	0.001	Plot width as a fraction of frame width
YAXISL	475.	0.001	Plot height as a fraction frame height

The members of NAMELIST /GRAPH/ are read in SUBROUTINE GRAFSET, and stored in COMMON /GRAFIX/. The actual graphical output can be generated either during or after processing. The standard distribution of the plotting routines written specifically for ESME v. 2001 use pgplot.[15] A sample post-processor is described later in Appendix B. Users wishing to process ESME data independently or with different graphics routines may find it of some use.

The options enabled by the **O** command are largely self-explanatory; the default PLTSW settings result in the output of a phase space plot of the distribution and the bucket due to RF source 1, as well as plots of the projections of the distribution along the θ and E directions. An example of an **O** command requesting such output every 100 turns could be

```
$GRAPH    MPLOT=100 THPMIN=-10.0 THPMAX=10.0
          DEPMIN=-25.0 DEPMAX=25.0 $END
```

in which the limits are appropriate ones chosen by the user. In addition to the graphical output generated as a result of the O command, the first and second moments of the distribution are computed and output, as well as a number of other system parameters. The moments included on the plots are derived from the particles which are in the class(es) being plotted and within the plot limits. The moments printed in the standard output are those for the entire distribution. The default limits for many of the plots serve as flags to the plotting routine to choose reasonable limits. Note that the limits of the phase space projection plot (THBMIN, EBMIN, *etc.*) are necessary only if they are different from the phase space plot limits. The parameter DELCON is included to allow the user to either determine the separatrix arbitrarily closely or to save processing time, since in certain situations the routine which determines the bucket in ESME is required to perform many iterations of the difference equations in order to determine the separatrix to the specified accuracy. In those instances in which the contour-drawing routine is unacceptably slow (or unable) to find a contour, it may be useful to set KNTLIM to some lower number. The default values (XCRNR, YCRNR, XAXISL, and YAXISL) give a satisfactory result for landscape orientation on an 8.5 × 11 sheet, but may need to be adjusted for other devices.

2.2.5 T Command - Track distribution

<table>
<tr><th colspan="5">T Command, Namelist /CYCLE/</th></tr>
<tr><th rowspan="2">Variable</th><th colspan="2">Default</th><th rowspan="2" colspan="2">Description</th></tr>
<tr><th>Value</th><th>Unit</th></tr>
<tr><td>TSTOP[a]</td><td>0.0</td><td>s</td><td colspan="2">Time at which to stop tracking</td></tr>
<tr><td>TTRACK[b]</td><td>0.0</td><td>s</td><td colspan="2">Duration of time to track</td></tr>
<tr><td>MSTEP</td><td>100</td><td>-</td><td colspan="2">Number of tracking steps (minimum) per synchrotron period</td></tr>
<tr><td>RSCALE0</td><td>1.0</td><td>-</td><td colspan="2">Ratio of time per iteration to circulation period τ [c] — See text!</td></tr>
<tr><td>LGRTHM</td><td>1</td><td>-</td><td colspan="2">Select difference equations used in tracking
1–Complete kinematics, expand path length to maximum order using input coefficients ALPHAn[d]
2–Use the simplified difference equation
$\vartheta_{i,n} = \frac{\tau_{s,n}}{\tau_{s,n-1}}\vartheta_{i,n-1} + 2\pi\eta\frac{\Delta p}{p}$</td></tr>
<tr><td>ITRAP(1:4)</td><td>0</td><td>-</td><td colspan="2">Indicates a condition for which tracking should be interrupted before time indicated by TTRACK or TSTOP:
0–No trap
1–Trap on minimum bunch width
2–Trap on minimum bunch height
3–Trap for $\eta = $ ETATRP (tolerance $\Delta\eta/\eta = \pm.01$)
4–Trap for $|\phi_s| = $ PHISTRP (tolerance $\Delta\phi_s = \pm.005$)
5–Trap for $\eta > 0$ (transition crossing)
10 – 19–Call SUBROUTINE SHAZAM, enter at SHAZAM, SHAZAM1, SHAZAM2, … following every iteration of the difference equations
-1 – -10–Call SUBROUTINE SHAZAM, enter at SHAZAM, SHAZAM1, SHAZAM2, … before every iteration of the difference equations</td></tr>
<tr><td>ETATRP</td><td>.001</td><td>-</td><td colspan="2">For ITRAP = 3; tracking stopped when $\eta = $ ETATRP</td></tr>
<tr><td>PHISTRP</td><td>.95</td><td>-</td><td colspan="2">For ITRAP = 4; tracking stopped when $|\sin\phi_s| = $ PHISTRP</td></tr>
<tr><td>MGRACE</td><td>0</td><td>-</td><td colspan="2">Allow a "grace period" of MGRACE turns before trapping conditions are checked</td></tr>
<tr><td>HISTRY</td><td>F</td><td>-</td><td colspan="2">Write history records to logical unit 9 during tracking[e]</td></tr>
<tr><td>HISTSIZ</td><td>10^4</td><td>-</td><td colspan="2">Approximate number of history records for one execution of T</td></tr>
<tr><td>BBDRY</td><td>F</td><td>-</td><td colspan="2">Remove particles tracked outside of region $-180°/\text{FRAC} \le \theta \le 180°/\text{FRAC}$</td></tr>
</table>

[a]TSTOP is set to 0.0 when tracking is completed, or interrupted by an ITRAP option.

[b]TSTOP takes precedence; if TSTOP=0.0, then TTRACK determines duration of tracking.

[c]Scaling with RSCALE0 > 1 limited by MSTEP

[d]See R command

[e]See Section 3.2

The members of NAMELIST /CYCLE/ are read in SUBROUTINE CYCPROG and stored in COMMON /CYCLP/. The options in the **T** command direct the flow of processing in SUBROUTINE CYCPROG, which iterates the difference equations for the particles. A typical **T** command line might be

$CYCLE TTRACK=.001 LGRTHM=2 HISTRY=T $END

in which tracking with accumulation of a history file is specified for .001 seconds of simulated time using the simplified form of the difference equations. Note that *either* TSTOP or TTRACK may be used to specify the duration of tracking, though TSTOP takes precedence. Also, since an interruption in tracking by an ITRAP option (for ITRAP < 6) results in TSTOP being set equal to 0.0, the stop time must be re-specified in a subsequent **T** command. Any condition which halts or interrupts tracking is checked at most once per titeration, so tracking duration may be slightly longer than specified by TSTOP or TTRACK. Four ITRAP variables are provided to allow for multiple traps and/or calls to SHAZAM routines during tracking. If $10 \le \text{ITRAP} \le 19$, then a call is made to the corresponding SHAZAM entry point following every turn (or RSCALE turns, if RSCALE0 $\ne$ 1). Similarly, for $-19 \le \text{ITRAP} < -10$ a SHAZAM call is made at the beginning of an iteration, immediately preceding the loop over individual macroparticles. Tracking is not interrupted by these calls unless the SHAZAM routine itself fails to make a normal return. The ITRAP values $1 - 7$ may also be of either positive or negative sign with the effect that the trap condition is checked respectively after or before the iteration of the map, a choice that will only rarely have significance.

The parameter RSCALE0 is extremely effective in reducing calculation time in large scale multiparticle models.[1] The effect is a factor of RSCALE**-4 in time used for a rather wide class of problems. See the reference cited for the concept and limitations. An appropriate choice can be two or greater; RSCALE0 (input datum) is limited to RSCALE (program parameter) according to MSTEP to prevent extreme over-scaling. If nothing is done but setting RSCALE0, the calculation time is cut by 1/RSCALE. The further gains require reducing the number of macroparticles, a subject which is treated in the reference.

2.2.6   Q Command - Quit

The **Q** command directs ESME to cease processing. No more commands are read.

2.2.7   D Command - Display

The **D** command directs ESME to generate graphical output at the point at which the command is issued. The form of the output is specified by the most recent **O** command. The **D** command is useful for generating output at a particular point in a calculation, since the **O** command itself only provides output every MPLOT turns.

2.2.8   W Command - Write comment

The **W** command simply directs the program to echo the characters following W (on the same line and after four spaces) to the standard output. It is intended to provide the user with the ability to insert comments into an input dataset which appear as comments in the output file also.

2.2.9 H Command - History output

<table>
<tr><td colspan="5" align="center">H Command, Namelist /HISTRY/</td></tr>
<tr><td rowspan="2">Variable</td><td colspan="2" align="center">Default</td><td rowspan="2" colspan="2" align="center">Description</td></tr>
<tr><td>Value</td><td>Unit</td></tr>
<tr><td>IDEV</td><td>1</td><td>-</td><td colspan="2">The virtual device number for output</td></tr>
<tr><td>NPLT(1:2,1:50)</td><td>0[a]</td><td>-</td><td colspan="2">Index of element in history records; NPLT(1,I) is independent variable, NPLT(2,I) is dependent variable
Real records:
1 – Time
2 – PHIS, the synchronous phase on same scale as ϑ[b]
3 – PDOT, dp_s/dt
4 – THBAR, the mean value of θ for the distribution
5 – EBAR, the average energy of the distribution
6 – THRMS, the rms spread in θ of the particles
7 – ERMS, the rms energy spread
8 – ES, the synchronous energy
9 – E0, the energy on the reference orbit
10 – ES-E0
11 – THREF, the azimuth of a particle tracked from (0,ES)
12 – EREF, the energy of a particle tracked from (0,ES)
13 – EPSILON, the emittance[c]
14 – NUS, the synchrotron frequency
15 – SBCKT, the RF "bucket" area
16 – HBCKT, the RF "bucket" height
17 – ETA, $\gamma_T^{-2} - \gamma^{-2}$
18 – RSCALE; see **T** command
19 – TAU, synchronous revolution period
20 – PSIADD, phase feedback; see **L** command
21 – DAMPL, voltage feedback factor; see **L** command
22 – DELR, synchronous orbit radius - reference orbit radius
Integer records:
31 – ITERATION NUMBER
32 – KNTSC, number of particles in $\varepsilon_\ell \neq 0$ partition of distribution</td></tr>
<tr><td></td><td></td><td></td><td colspan="2">Array records:
 51- 60 – SPARE(1-10)
101-110 – EV(1-10)
111-120 – PSI(1-10)
121-130 – FREQ(I)-FRI(I), Change in frequency of source I
201-250 – FAMPL(1-50), Fourier amplitudes; see **F** command
251-300 – FAZE(1-50), Fourier phases; see **F** command</td></tr>
<tr><td>NWRT(1:2,1:50)</td><td>0[a]</td><td>-</td><td colspan="2">Indices of elements in history records; works like NPLT but writes records to FORTRAN UNIT 19</td></tr>
<tr><td>IMETA</td><td>0</td><td>-</td><td colspan="2">Directs graphic output
< 0 - Output written to HIGZ metafile
 0 - Output to device selected by IDEV
> 0 - Output to both metafile and selected IDEV</td></tr>
<tr><td>XCRNR</td><td>135.</td><td>0.001</td><td colspan="2">Fraction of full width of plot frame between lefthand edge and left side of plot</td></tr>
<tr><td>YCRNR</td><td>69.0</td><td>0.001</td><td colspan="2">Fraction of full height of plot frame between bottom edge and bottom of plot</td></tr>
<tr><td>XAXISL</td><td>750.</td><td>0.001</td><td colspan="2">Plot width as a fraction of frame width</td></tr>
<tr><td>YAXISL</td><td>475.</td><td>0.001</td><td colspan="2">Plot height as a fraction frame height</td></tr>
</table>

[a]The default value of 0 indicates to SUBROUTINE HISTORY that all of the desired history plots have been generated, so only the first set of consecutive nonzero entries to array NPLT or NWRT will select data.

[b]See **A** command description.

[c]EPSILON $= \text{ANORM } \tau(\sqrt{\sum \theta_i^2 \sum E_i^2 - (\sum \theta_i E_i)^2})/N$ eVs; see text for ANORM.

The members of NAMELIST /HISTRY/ are read in subroutine HISTORY. The emittance EPSILON logged by the **H** command is the rms area in eVs. There is a switch RENORM for the **P** command which controls whether a factor ANORM is included in the emittance. If RENORM is .TRUE., ANORM is included. It is calculated by the P command to scale the rms emittance of the initial bunch to the area SBNCH of the curve used to generate it. If the intial distribution is not a matched bunch, setting RENORM will have no effect on the calculated emittance. Such scaling is not a good idea in general because the factor is calculated only for the initial distribution.

The output produced by the **H** command, as implied by the simplicity of the /HISTRY/ NAMELIST, is not as flexible as the output produced by the **O** command. The user simply specifies pairs of values to be plotted using the indices above in array NPLT, and optionally the device number, and the program produces a history plot for each pair of inputs assembled sequentially from the entire history record (*i.e.*, there is no choice of a range for any axis). For example, suppose that a simulation has run for 10,000 turns, and the desired output is a record of the distribution moments *vs.* time over that period. Then a suitable **H** command is

$HISTRY NPLT=1,4,1,5,1,6,1,7,4,5 $END

which will generate plots of THBAR, EBAR, THRMS, and ERMS *vs.* time, as well as a plot of EBAR *vs.* THBAR. The fortran array index convention is exploited to avoid explicit reference to the indices; this is the recommended manner of input.

Because the frequency of write operations to the history tape may be modified by the program and because of a culling procedure applied to the data points by the plotting routine, not every turn will be included in the plot for very long simulations. The frequency of the records written to file is governed by the HISTSIZ parameter of the **T** command which defaults to 10^4. With this value the tracking routine will write a record every turn if there are fewer than 10^4 iterations between the time limits given in the **T** command. Because there can be a few hundred four byte words per record, this file can easily be a few MB per **T** command. Whether HISTORY *plots* every record depends on the space available for the plot. This can be increased by the KMAXPTS parameter of the **Y** command. For more elaborate plots or, perhaps, for analysis, anything which can be plotted according to NPLT values can be written to logical unit 19 using NWRT.

2.2.10   M & N Commands - Save mountain range data & Plot mountain range

<table>
<tr><td colspan="5" align="center">M Command, Namelist /MRANGE/</td></tr>
<tr><td></td><td colspan="2" align="center">Default</td><td></td></tr>
<tr><td>Variable</td><td>Value</td><td>Unit</td><td>Description</td></tr>
<tr><td>IPU</td><td>.TRUE.</td><td>-</td><td>Set display to beam current</td></tr>
<tr><td>FASPEC</td><td>.FALSE.</td><td>-</td><td>Set display to fourier amplitude spectrum</td></tr>
<tr><td>FPSPEC</td><td>.FALSE.</td><td>-</td><td>Set display to fourier power spectrum</td></tr>
<tr><td>TMBEGIN</td><td>0.0</td><td>s</td><td>Time at which to start saving mountain range data</td></tr>
<tr><td>TMEND</td><td>0.0</td><td>s</td><td>Time after which to stop saving mountain range data</td></tr>
<tr><td>MRMPLOT</td><td>1</td><td>turn</td><td>Turn interval at which to record mountain range data</td></tr>
<tr><td>MRNBIN</td><td>100</td><td>-</td><td>Number of bins in plot range MRBMAX - MRBMIN</td></tr>
<tr><td>MRBMIN</td><td colspan="2" align="center">[a]</td><td>Minimum ordinate for mountain range</td></tr>
<tr><td>MRBMAX</td><td colspan="2" align="center">[b]</td><td>Maximum ordinate for mountain range</td></tr>
</table>

[a]Defaults to $-180°$/FRAC for current pickup data and to one for fourier amplitude or power spectrum.

[b]Defaults to $180°$/FRAC for current pickup data and NBINFFT/2 for fourier amplitude or power spectrum.

<table>
<tr><td colspan="4" align="center">N Command, Namelist /MRPLOT/</td></tr>
<tr><td></td><td colspan="2" align="center">Default</td><td></td></tr>
<tr><td>Variable</td><td>Value</td><td>Unit</td><td>Description</td></tr>
<tr><td>MRPMIN</td><td>0.0[a]</td><td>-</td><td>Minimum abscissa value for mountain range plot</td></tr>
<tr><td>MRPMAX</td><td>0.0[a]</td><td>-</td><td>Maximum abscissa value for mountain range plot</td></tr>
<tr><td>NTRACE</td><td>100</td><td>-</td><td>Number of traces on page</td></tr>
<tr><td>NSKIP</td><td>0</td><td>-</td><td>Number of records to be skipped between each trace</td></tr>
<tr><td>TOPTOB</td><td>0.7</td><td>-</td><td>The fraction of the vertical range over which NTRACE traces are to be plotted (approximate if TBASE=T)</td></tr>
<tr><td>SCALE</td><td>0.3</td><td>-</td><td>The height of the first trace, in units in which the entire vertical range of the plot is 1.0</td></tr>
<tr><td>MSTART</td><td>0[b]</td><td>-</td><td>Iteration number at which to start plots</td></tr>
<tr><td>MSTOP</td><td>0[b]</td><td>-</td><td>Turn number at which to stop plots</td></tr>
<tr><td>TMSTART</td><td>0.0[b]</td><td>s</td><td>Time at which to start plots</td></tr>
<tr><td>TMSTOP</td><td>0.0[b]</td><td>s</td><td>Time at which to stop plots</td></tr>
<tr><td>TBASE</td><td>F</td><td>-</td><td>Switch causing plot trace separation to be proportional to time</td></tr>
<tr><td>NRNBIN</td><td>100</td><td>-</td><td>Number of points to plot on a trace</td></tr>
<tr><td>SMOOTH</td><td>0</td><td>-</td><td>Smoothing option
-1 — 1-2-1 averaging of adjacent bins
 0 — No smoothing
 1 — Bernstein polynomial smoothing</td></tr>
<tr><td>OBJWGT</td><td>0.05</td><td>-</td><td>Weight of fitting term of object function w/ smoothing term for polynomial smoothing</td></tr>
<tr><td>ITNO</td><td>1</td><td>-</td><td>Number of iterations for either 1-2-1 or Bernstein smoothing</td></tr>
<tr><td>LIM</td><td>F</td><td>-</td><td>Switch for plotting dotted lines connecting left-most and rightmost non-zero points of consecutive traces[c]</td></tr>
</table>

[a]Defaults of 0.0 for MRPMIN and MRPMAX imply data is to be plotted over its entire range; abscissa units are degrees for IPU=T and harmonic number for FAxSPEC=T.

[b]The defaults of 0 for MSTART and MSTOP, or 0.0 for TMSTART and TMSTOP, imply that all mountain range records are to be plotted

[c]SMOOTH = 1 also required.

N Command, Namelist /MRPLOT/, continued			
Variable	Default Value	Unit	Description
IDEV	1	-	Output device
IMETA	0	-	Directs graphic output < 0 - Output written to HIGZ metafile 0 - Output to device selected by IDEV > 0 - Output to both metafile and selected IDEV
XCRNR	135.	0.001	Fraction of full width of plot frame between lefthand edge and left side of plot
YCRNR	69.0	0.001	Fraction of full height of plot frame between bottom edge and bottom of plot
XAXISL	750.	0.001	Plot width as a fraction of frame width
YAXISL	475.	0.001	Plot height as a fraction frame height

The members of NAMELIST /MRANGE/ are read in subroutine MRINIT and stored in the common block /MRANGE/ in response to the **M** command. The members of NAMELIST /MRPLOT/ are read in the subroutine MRPLT in response to the **N** command and also stored in /MRANGE/.

The **M** and **N** commands are intended to provide plots similar to the display provided by an oscilloscope recording successive traces from a beam current pickup, each trace vertically displaced from the previous one. The default display depicts the time evolution of the azimuthal projection of a distribution in a manner which somewhat resembles a mountain range, hence the name. The **M** command directs the program to save the data, while the **N** command directs the program to process the data which have been saved in a file and produce mountain range plots. The **M-N** combination can also be used to plot the evolution of the fourier amplitude or power spectrum of the beam current. Only one type of plot can be produced at a time, so only one of the switches IPU, FASPEC, FPSPEC should be set .TRUE. . The setup of the desired range of fourier components is handled through either the **F** or **B** commands. If there is no need for the fourier analysis other than the mountain range plot, *e. g.* no calculation of the voltage arising from the beam image current or bunch self field, then the value of MFFT in the FFT namelist should be set to the number of turns between traces MRMPLOT to save on unnecessary evaluations of the fft. A typical pair of **M** and **N** commands might be

> $MRANGE TMBEGIN=0.0 TMEND=1.0 MRMPLOT=1000 $END

in which mountain range records are recorded every 1000 turns of tracking from 0.0 to 1.0 seconds, and

> $MRPLOT SMOOTH=1 $END

which directs that all of the data accumulated thus far be plotted in mountain range format with Bernstein polynomial smoothing. The default for TBASE (F) results in a fixed vertical separation between consecutive traces. If TBASE = T, the vertical separation between consecutive traces will be proportional to the time separating their records, better simulating the mountain ranges normally depicted on an oscilloscope.[3] The values for TOPTOB and/or SCALE may have to be adjusted to achieve a satisfactory effect.

If MRNBIN $\neq$ NRNBIN, the saved data is transformed to the correct number of bins for plotting by cubic spline interpolation. This feature is independent of whether the stored distribution is first smoothed. Either Bernstein polynomial smoothing[16] (SMOOTH = 1) or 1-2-1 averaging of adjacent bins (SMOOTH = -1) may be used. The spline interpolation can produce some smoothing depending on the particular NRNBIN and MRNBIN values. The defaults for MRNBIN and NRNBIN will usually give very smooth plots with SMOOTH = 1, but the default value of 100 for MRNBIN may be too large for a sparsely populated distribution if SMOOTH = 0. Either type of smoothing may be iterated by raising ITNO. The two techniques are quite different, however, and the meaning of iteration is rather different. The Bernstein polynomial smoothing is a global procedure which seeks to minimize an object function with contributions from both a smoothness measure and a squared fitting error. Iteration attempts repeated minimizations of this object function. The relative weight given to the fit is increased by increasing OBJWGT; the default of 0.05 gives considerable weight to fitting. The 1-2-1 smoothing is a local weighted averaging. Iterating it broadens the span of the averaging giving 1-4-6-4-1, 1-6-15-20-15-6-1, ...smoothing for ITNO = 2,3, ...respectively. (The relative weights are $C_i^{2 \cdot \mathrm{ITNO}}$ for $i = 1, \ldots, 2 \cdot \mathrm{ITNO}$.) Generally some smoothing will be required to make mountain range plots that look like those made from an accelerator beam current pickup using an oscilloscope. Typical choices are SMOOTH=-1,ITNO=2 and SMOOTH=1,ITNO=1.

---

[3]In relativistic situations, no difference will be discerned between the plots generated with TBASE either T or F.

2.2.11 <u>L</u> Command - Low level feedback parameters

Variable	Default Value	Unit	Description
PHFBON	F	-	Activates phase feedback
NTUAVG	1	-	The number of past turns to average in computing phase feedback; the default NTUAVG = 1 corresponds to infinite-bandwidth.
NTURES	1	-	The number of turns for the feedback to respond; the present signal is compared to the signal of NTURES turns ago.
ITFB	0	-	The form of phase feedback: 0 – Critical damping 1 – Critical damping; gain reduced proportionally to ν^2 within non-adiabatic interval 2 – Fixed 3 – Fixed; gain reduced proportionally to ν^2 within non-adiabatic interval
FBFACT	1.0	-	The gain applied to the phase feedback
USEWT	F	-	Applies weight function W to phase signal over NTUAVG turns
W(1:NTUAVG)	0.0	-	Weight function multiplying phase signal
DLIMIT	5.7296	deg	The upper limit on the magnitude of the phase feedback on a given turn
VFBON	F	-	Activates voltage feedback
VFBFCTR	1.0	-	The gain applied to the voltage feedback
VLIMIT	.1	-	Limit on the fraction of voltage feedback (DAMPL)
FILBWD	'DUMMY'	-	Name of time/bunch length curve for voltage feedback (see text)
ETAJMP	0.0	-	The value of η^a at which to jump the phase of the RF
BLFBON	F	-	Activates fundamental beam loading feedback correction
BLFFON	F	-	Activates fundamental beam loading feed-forward correction
ICAV	1	-	Index of the rf system to use in beam loading calculation
NHALF	-	-	Number of bins spanning half a bucket in the longitudinal charge distribution
RSHUNT	-	ohm	Loaded shunt resistance of cavity(ies) for beam loading calculation
QCAV	-	-	Loaded Q of cavity(ies) for beam loading calculation
DTUNE	0.0	-	$\Delta f/f$ detuning of cavity(ies) in beam loading calculation
ENQ	2.D10	-	Number of beam particles represented by entire macroparticle distribution [b]
GAIN	0.0	-	Loop gain for feedback correction of beam loading
DELAY	-	s	Loop delay in feedback correction to beam loading
FTOP	-	Hz	Upper cutoff for beam loading correction feedback loop
WDTHFCTR	1.0	-	Ratio of feed-forward current pulse widths to bunch rms widths
CHGFCTR	1.0	-	Ratio of feed-forward current pulse charges to bunch charges

L Command, Namelist /LLRF/

[a] $\eta = \gamma_T^{-2} - \gamma^{-2}$

[b] Can be read with **L** or **B**

The members of NAMELIST /LLRF/ are read in subroutine LOWLVL and stored in COMMON /FEEDS/. The phase feedback, intended primarily to damp dipole bunch oscillations, is computed according to the following formula for ITFB = 0:

$$\mathrm{PSIADD} = \frac{\mathrm{FBFACT} \sum_{i=1}^{\mathrm{NTUAVG}}(W_i(\bar{\theta}_{n-i} - \bar{\theta}_{n-\mathrm{NTURES}-i})}{\pi\nu_s\mathrm{NTUAVG}^2 \sum_{i=1}^{\mathrm{NTUAVG}} W_i}$$

where n is the current turn number, and ν_s is the synchrotron frequency. For the option ITFB = 1 the gain is reduced within the non-adiabatic interval around transition proportionally to ν^2. For ITFB = 3, ν_s is removed from the denominator of the feedback formula above giving a constant-gain feedback, and FBFACT should be reduced to some appropriate value like $4.0 \cdot 10^{-3}$ for example. For ITFB = 3 the constant-gain formula is is reduced in the non-adiabatic interval.

A simple invocation of phase feedback would appear as

$LLRF PHFBON=T $END

The defaults imply critical damping. The routine automatically reduces the gain within the non-adiabatic time around transition proportionally to ν_s^2; if the feedback gain is an object of study, this function of the routine LOWLVL can be managed by a user-written SHAZAM.

The voltage feedback, intended to damp quadrupole bunch oscillations, operates according to

$$V'_n = (1 + (\text{VFBFCTR} \times (\sigma_{\vartheta,n} - \sigma_{\vartheta,n-1} - \varsigma_{\vartheta,n} + \varsigma_{\vartheta,n-1})/\nu_s) \times V_{n-1}$$

where n is the current turn number, σ is rms bunch width, ς is the program value for bunch width, ν_s is the synchrotron tune, and VFBFCTR is an arbitrary gain. VLIMIT is the maximum permitted for the ratio V'_n/V_n. This ratio is the variable DAMPL in /FEEDS/, and it may be plotted if desired by using a SHAZAM entry to store it as one of the SPARE variables and using the **H**istory command to plot it. If VFBFCTR is $< 0.$, the feedback may be used to produce a controlled emittance dilution. If the FILBWD variable is not read in to point to a bunch length curve, the ς values are taken as $\sigma_{\vartheta,1}$ for all time. This default works fine for a constant fixed bucket or over a period in which there is little secular change in the bunch width. The use of a target curve for the bunch width is a way of avoiding the difficulty of representing the correct bandpass frequency response in a time domain calculation. It is in a certain sense an idealized feedback algorithm, but in a programming sense it is something of a kluge. If one has a curve generating program to produce voltage curves, it can be used to produce the corresponding bunch length curve for a chosen longitudinal emittance. The format for a bunch width curve is a first line containing an integer specifying the number of entries to follow with the following entries consisting of two numbers per line giving a time in seconds and a bunch width at that time. The table is limited to 350 or fewer entries. (parameter LTABL in parameters.inc) The program normalizes the curve so that the first time slot has the value of σ_ϑ at the time of the **L** command. Obviously if the curve is not read in at the time corresponding to the initial point in the table, this could cause a scale error. The default feedback gain is arbitrary; it may be necessary to adjust VFBFCTR to obtain satisfactory results. All of this may seem a little awkward and *ad hoc*, but it really works very well for reducing shape oscillations after transition, for example.

The LOWLVL routine can model beam loading feed-forward or feedback correction. The azimuthal current distribution can be fed back with a given loop gain (GAIN), frequency bandwidth ($h\omega_\circ$ to FTOP), and loop delay (DELAY). The cavity can be detuned by $\Delta\omega/\omega$ (DTUNE), or the detuning corresponding to the equilibrium correction for beam loading can be calculated (DTUNE > 1). The beam intensity is given by ENQUE, the total number of particles represented by the macroparticle distribution. One rf system, selected by ICAV, can be modeled; the effective shunt resistance (RSHUNT) and Q (QCAV) characterize the cavity(ies). To avoid details of the low level system, the feedback is treated entirely in terms of the the equivalent generator current it introduces. Delay means shifting the correcting current in time and frequency response of the loop means a weighted average over the appropriate number of earlier time bins in the charge distribution. At present, the weighting is simply a linear decrease in weight from the bin at time present to the bin at time $(\text{FTOP})^{-1}$ earlier. At the time of writing (September 8, 2000) the beamloading feedback model is being refined. What is available has been useful, but there may be substantial changes in this particular feature over a rather short time scale.

Feed-forward feeds to the cavity a series of current pulses at the times of the bunch centroids with width proportional to the rms bunch width (proportionality factor WDTHFCTR) and charge proportional to the bunch charge (proportionality factor CHGFCTR).

Only fundamental beam loading has been taken into account. The effects of higher order modes without correction can be modeled using the time domain resonance treatment of the **B** command. To include changes which occur during the acceleration cycle in the properties of the cavity other than the fundamental frequency requires additional **L** commands.

2.2.12 <u>B</u> Command - Beam-derived potential (self-force and $Z_{||}$)

Variable	Default Value	Unit	Description
	Default		
Variable	Value	Unit	Description
A	0.005	m	Effective beam radius
B	0.05	m	Effective beampipe radius
ENQ	$2 \cdot 10^{10}$	-	Number of protons to be represented by the distribution
NBINSC	100	-	Number of bins for histogram of charge distribution
MSC	1	-	Collective effects are to be calculated MSC times between rf cavities[a]
TCHGON	0.	s	End of period starting at TIME = 0. in which beam charge is ramped linearly from 0. to ENQ
SCON	F	-	Activate time domain perf. cond. wall calculation
TDON	F	-	Activate time domain wake field calculation with supplied response function[b]
FDON	F	-	Activate frequency domain wall impedance calculation
FDSCON	F	-	Activate frequency domain calculation of perf. cond. wall voltage
QREZON	F	-	Activate time domain calculation for high-Q resonance
NBINFFT	256	-	Number of bins to be used in Fourier transform
MFFT	1	-	Interval (in turns) between Fourier transforms
NNF [c]	0	-	Number of fourier harmonics to be stored in history
NF(1:NNF)[c]	0	-	Harmonic numbers of fourier spectrum components to be stored
NBRES	1000	-	Number of time slices for time domain solution of high-Q resonator
NIXNOIS	0	-	Three-way switch to control smoothing of charge distribution -1 => 1-2-1 averaging of adjacent bins 0 => no smoothing measures 1 => Bernstein polynomial smoothing
ITKNT	1	-	Number of iterations for either 1-2-1 or Bernstein smoothing
OBWGT	0.05	-	Weight of fitting vs. smoothing in object function for Bernstein smoothing
FILIMP	'DUMMY'	-	Full path for file containing the impedance table
FILRES	'DUMMY'	-	Full path for file containing the resonance list
FILTDB	'DUMMY'	-	Full path for file containing time domain response basis

[a]The number of such calculations per turn will be MSC*NCAV; NCAV is a parameter of the **R** command.

[b]see Appendix 3.2.6

[c]For instruction in the use of NF(1:NNF), see the description following the **F** command.

The members of NAMELIST /SCHG/ are read in subroutine BEAMSC and stored in COMMON /SPCHG/. The **B** command controls facilities in ESME for modeling the interactions of the beam particles with each other through both the direct particle-particle force and through wake fields excited as a consequence of the interaction of the beam with its environment (vacuum chamber, rf cavities, *etc*...). Both types of collective voltage can be calculated in either time domain or frequency domain. For the perfectly conducting wall term either is convenient and the difference is whether the gradient of the charge distribution is calculated from a local derivative or a fourier fit to the entire distribution. If frequency domain is used for the wall external impedance, it is much faster to use the fft machinery for the perfectly conducting wall also; set FDON=t for the impedance calculation and FDSCON for the perfectly conducting wall. If a response to a unit charge triangular pulse is available, one can use TDON=T; the time domain perfectly conducting wall calculation is selected by SCON=T.If the longitudinal impedance is representable by a few resonances, a time domain calculation can be made using a time domain calculation selected by QREZON=T, which is most accurate for high-Q resonances. Because time domain and frequency domain can be used together, low-Q resonances can be treated in frequency domain. Since the transients from low-Q impedances decay rather quickly, the impedance approach is likely to be appropriate for them

QREZON will apply to the calculation of the voltage due to those resonators for which the multipliers (third parameter in the resonance file record) are zero. The calculation is made in time domain and is based on a Green's function for a simple resonance. If there is one physical resonator in the ring, the periodicity in the problem is one. Thus, the FRAC parameter for the **R** command should be one. If the resonance is associated with the rf cavities, and those are distributed (more or less) uniformly around the ring in NCAV groups, then FRAC = NCAV is the proper periodicity. (NCAV is also a parameter for the **R** command.) Clearly, for an accelerator with high harmonic number, the calculation may be impracticably slow. The parameter NBRES must be large enough to give several bins per bunch width even to get the lowest mode of bunch-to-bunch coupling accurately. However, the time domain machinery is efficient, and it possible to run calculations with large NBRES like 10^5, for a somewhat extreme

example.

The more general version of the time domain calculation of the voltage arising from the beam image current flowing in $Z_\parallel$ requires that the user provide a file containing the values of a basis function at a regular tabular interval. The basis function is the response of the impedance to a unit current moving with the beam velocity distributed in a triangular pulse with base width twice the the tabular interval.[4] Except for special cases like a simple resonance, it will be necessary to calculate this response using a time domain EM program like TCBI. The first record in the basis function file FILTDB is the time length of the interval between evaluations. Following are the tabular values of the basis function continuing to times as long as the wake field is significant, or up to τ/FRAC if that is shorter. This calculation does not share with the high-Q resonator calculation the facility for accumulating excitation over multiple turns. If there is appreciable wake field extending beyond τ/FRAC, its effect is lost. However, something with so high a Q is probably correctly represented as a resonance or a few resonances and can be properly treated with the time domain resonance calculation (QREZON = .TRUE.). All three types of collective voltage calculation can be carried on simultaneously for different parts of the impedance.

One of the frustrations in calculating emittance growth caused by the collective potential is spurious emittance growth of an initial bunch matched to a single particle trajectory. It is generally difficult to produce a self-consistent distribution directly. Two ways are provided to avoid the initial match problem. The parameter TCHGON can be set to a value of a few synchrotron periods. The strength of the beam charge will then be ramped linearly to its target ENQ by the time TCHGON, thereby approximating an adiabatic introduction of the perturbation. Another technique which may require less computing is to generate a collective potential using the **B** command and an approximate initial distribution then to use the **K** command to remove the distribution before using the **P** again to provide a bunch matched to the new potential. This cycle could be iterated for an exacting application to obtain an adequate approximation to self-consistency.

Under some conditions, most likely at low energies, the collective potential is very significant, and applying it only once per turn is not realistic because there is too much force-free drift between energy increments. A typical symptom of this problem is spurious bunch breakup. The parameter MSC can be set to apply the collective potential MSC $\times$ NCAV times per turn, where NCAV is the number of cavity locations in the ring set by the **R** command. However, the collective potential itself changes very little in one turn, so it is evaluated once per turn independently of MSC and NCAV.

An example of **B** command which includes the self-force, resistive wall impedance, and some resonant elements is

> $SCHG FDON=T FDSCON=T QREZON=T A=.01 B=.03 ENQ=2.4E10 NBINSC=1000 NBINFFT=512
> NBRES=1000 FILIMP='IMPEDANCE.DAT' FILRES='RESONANCE.DAT' $END

in which the real and imaginary parts of the wall impedance are read from IMPEDANCE.DAT while the resonance values are obtained from the file RESONANCE.DAT. Note that the number of protons specified is for $1/\mathrm{FRAC}$ of the machine circumference. The number of bins used in the FFT will be the lesser of NBINFFT or the storage allocation IFFT. The parameter NBINSC should be chosen so that there are an adequate number of bins over the width of the bunch(es) to provide good shape information. With these switches, the bunch self-field is calculated from the FFT of the charge distribution, the general longitudinal impedance is treated in frequency domain, and the resonant contributions are evaluated in frequency domain or time domain according to the value of the "multiplier" entry for each resonance.

There is no necessary numerical relation between NBINSC and NBINFFT; however taking them equal saves some interpolation. If the two are unequal, spline interpolation on a charge distribution of NBINSC bins is used to make a distribution of NBINFFT bins for the FFT. Two types of smoothing may be applied to the distribution of NBINSC bins. SMOOTH=-1 causes a local averaging with weights 1/4, 1/2, 1/4. If ITKNT is greater than one, the smoothing is repeated effectively producing (less local) averaging with, for example, weights 1/16, 1/4, 3/8, 1/4, 1/16 for ITKNT=2. If SMOOTH=1, Bernstein polynomial smoothing[16] is employed. This is a global technique in which an object function containing a least squares fitting criterion and a smoothness measure is minimized. For this case ITKNT controls the maximum number of attempts to minimize the object function. This method is limited by a requirement that NBINSC be small enough that each bunch are spanned by fewer than 100 bins and, if there are multiple bunches, that there be three or more empty bins between them.

Another major hurdle in modelling high intensity beams is dealing with a sufficient number or macroparticles to keep local fluctuations in the space charge force from causing a spurious disruption of the distribution. When only the low frequency effects of the beam charge are of interest, it may be satisfactory to take a limited bandwidth in frequency domain and/or smooth the charge distribution using one of options of the SMOOTH control variable. The validity of these expedients is not always clear. A more rigorous alternative[1] is to set the parameter RSCALE0 of the **T** command to something larger than one; this option will reduce the number of iterations required by a factor $(\mathrm{RSCALE})^{-1}$ and permit reducing the number of bins by the same factor. For most problems, if the number of bins is so reduced, the number fo macroparticles can be reduced by $(\mathrm{RSCALE})^{-3}$.

---

[4]This approach is developed from the ideas of G. Sabbi,[14] but is not based on his code TRISIM.

2.2.13   F Command - Fourier transform

	Default		
Variable	Value	Unit	Description
FFTON	F	-	Activate Fourier transform calculation
FFTOUT	F	-	If TRUE, Fourier transform is printed
NBINFFT	256	-	Number of bins to be used in FFT
NNF	0	-	Number of fourier harmonics to be stored in history
NF(1:NNF)	0	-	Harmonic numbers of fourier spectrum components to be stored
MFFT	1	-	Frequency of Fourier transform calculation
NIXNOIS	0	-	Three-way switch to control smoothing of azimuthal histogram
			-1 => 1-2-1 averaging of adjacent bins
			0 => no smoothing measures
			1 => Bernstein polynomial smoothing
ITKNT	1	-	Number of iterations for either 1-2-1 or Bernstein smoothing
OBWGT	0.05	-	Weight of fitting vs. smoothing in object function for Bernstein smoothing

F Command, Namelist /FFT/

The members of NAMELIST /FFT/ are read in subroutine FOURFIT and stored in COMMON /FOURIR/. Many of the variables in NAMELIST /FFT/ are shared by NAMELIST /SCHG/, because the Fourier transform may be utilized in the collective potential calculation. The **F** command is intended to allow the user to examine the fourier transform of the distribution and to follow the development of selected fourier components. If, for example, one wants to track the turn-by-turn development of the first five odd fourier harmonics of the distribution, then the appropriate **F** command would be

> $FFT   FFTON=T NBINFFT=32 NNF=5 NF=1,3,5,7,9 $END

If at some point one should wish to plot the record for the amplitude of the third harmonic, then the **H** command should be issued to retrieve FAMPL(2).[5] The harmonics of the distribution should be multiplied by the periodicity in determining harmonics of the revolution frequency. If, for example, $FRAC = 7$, the fifth harmonic of the distribution corresponds to harmonic thirty-five of the beam circulation frequency. It is also possible to plot a time sequence of amplitude or power spectra as described in section 2.2.10.

The histogram can be smoothed before the FFT is performed. NIXNOIS = -1 and ITKNT = 1 results in a straightforward averaging of bins with their immediate neighbors using relative weights of 1-2-1. Increasing ITKNT results in higher order averaging. When NIXNOIS = 1, Bernstein polynomial smoothing is employed.[16] The relative importance attached to fitting of the data values and smoothness is governed by OBWGT; ITKNT determines how many trials are made to reduce the object function. An ITKNT of zero is valid; it results in a well-smoothed representation of the distribution but may remove detail which is meaningful. There are other remarks on smoothing in secs. 2.2.10 and 2.2.12 where mountain range plots and collective potential calculations are discussed.

---

[5]See Section 2.2.9, History command.

2.2.14　C Command - Flow contours

	Default		
Variable	Value	Unit	Description
LINES	2	-	The number of flow lines to be drawn
STVAL	0.,0.	a	The (ϑ, E) starting values for LINES different contours
STATIC	.TRUE.	-	Contours drawn with parameters fixed at initial values for STATIC = .TRUE.; parameters vary according to their programs for STATIC = .FALSE.
PLTBKT	.TRUE.	-	Option switch to include bucket contour on flow line plot
TSTOP^b	0.	s	End of interval over which maps are generated
TTRACK^c	0.	s	Time over which periodic flow maps will be generated
ACCEL0	1.0	-	Number of beam turns per step between mappings
PARTION	F	-	Switch indicates if points on separate flow lines belong to different partitions of the phase points
ITRAP(1:4)	1	-	Flags a condition for which the **C** command should be interrupted; see **T** command for values
ETATRP	.001	-	Trapping parameter; see **T** command
PHISTRP	.95	-	Trapping parameter; see **T** command
MGRACE	0	-	Trapping parameter; see **T** command

^{a}Starting values for contours have units degrees,MeV.

^{b}TSTOP set to 0. when calculation complete or interrupted by an ITRAP option.

^{c}TSTOP takes precedence; if TSTOP=0., TTRACK determines duration.

The **C** command is used to generate flow line maps at the plotting intervals MPLOT, which is set by the **O** command along with the other parameters controlling the graphics. It may also be used to establish a (zero-emittance) distribution to be tracked. The subroutine FLOWPRG reads the namelist /FLOW/ and stores some of the parameters in common /FLOWP/. However, FLOWPRG is a driver for many ESME subroutines so it distributes data among several common blocks.

When the **C** command is used to generate an initial distribution for tracking, TTRACK and TSTOP are set to zero so that only one set of points on flow lines is generated. It can be combined with a non-zero emittance distribution (*i.e.*, a bunch) if desired. Such a bunch can be generated before the flow lines and the resulting collective potential will then be taken into account in generating the flow lines if the **B** command has been called first. In any case, a zero-emittance type of distribution is stored in the array PHASE above a partition KNTSC so that only the bunch(es) affect the collective potential when that calculation is active. Furthermore, the flow line points are not used in calculating beam moments and emittance.

The **C** command differs from the **T** command in not tracking any distribution turn-by-turn and not calculating bunch properties or storing quantities for history plots. Because the **C** command works without a bunch-like ($\varepsilon_\ell \neq 0$) distribution, it does not ordinarily incorporate the effect of a collective potential. It is possible to use **C** to generate such flow lines by first populating a bunch (**P**) then invoking **B** and **C** in that order. Using the **K** command with KNTSET = -1 will remove the bunch if desired.

When the parameter STATIC is set false, the trajectories from the starting points STVAL(ϑ_i, E_i) will be generated over the time indicated by TTRACK or TSTOP. In this case quantities like rf voltage which have a time program will change as the trajectory advances. These dynamic trajectories are not lines of Hamiltonian flow, but they provide an alternative to fixed-time phasespace macroparticle distributions or fixed-time contours for visualizing the dynamics.

Static contours produced by the **C** command can also be produced by the ICONTUR = 5 option of the **O** command.

2.2.15 K Command - Kill all or parts of the distribution

Variable	Default Value	Unit	Description
			K Command, Namelist /KUTS/
KUT	0	-	Cut the last KUT particles from the distribution
KNTSET	0	-	Reset particle count $< 0 \Longrightarrow$ KNTSC $\longrightarrow 0$; removes bunch-type distribution $> 0 \Longrightarrow$ KOUNT $\longrightarrow 0$; removes entire distribution
K1	0	-	Starting point for a partial removal of the distribution
K2	[a]	-	End point for partial removal of distribution
KLASS	0	-	Selects a partition of the distribution to be removed[b]

[a] Defaults to KOUNT
[b] See PARTION in **P** command

The **K** command calls KARVE which reads the namelist /KUTS/ for integers which indicate what part of the current phase space distribution to kill. It is really a pastiche of several related functions which have surfaced at one or another time in SHAZAM routines (see **0 – 9** commands). Only one option can be exercised per call, and parameters are always reset to harmless before the command returns.

2.2.16 0 – 9 Commands - User-written SHAZAM routines

The SHAZAM facility is intended to allow users to integrate their own routines into the code. All COMMON blocks in ESME are available to SHAZAM. As indicated in section 2.2.5, SHAZAM entry points may be called before or following every iteration of the difference equations using the TRAP option provided through NAMELIST /CYCLP/. In addition, SHAZAM routines may be called explicitly using commands **0 – 9**. Of course, any subsequent input is up to the author of the code following the appropriate SHAZAM entry point. It may be NAMELIST-directed, as ESME largely is, or there may be no input at all. The user is cautioned, however, to ensure that any subsequent input data is read; otherwise ESME will read the line where it finds itself after returning from SHAZAM and attempt to interpret it as a command.

2.2.17 Y Command - Memory allocation for

	Y Command, Namelist /MEMORY/		
	Default		
Variable	Value	Unit	Description
KNPHASE	50001	-	The storage allocation for each phase coordinate — max. value for KOUNT +1
KMAXCVB	1000	-	The storage allocation for large tables of data or intermediate results in time domain collective voltage calculations [a]
KMAXPTS	1000	-	Storage allocation for history plots

[a] KMAXCVB sets the maximum allowable value for input parameters NBINSC and NBRES.

    The **Y** command allows one to allocate memory for arrays whose sizes vary over a wide range depending on the problem. There are moderate default levels (as indicated in the table above), so it may not be necessary to allocate for a problem with a few thousand macroparticles. The **Y** command calls the subroutine MEMALLOC which reads the namelist /MEMORY/ for integers which set the size of phase space coordinate storage, the plot storage for histories, and beam induced voltage distributions, *etc.*

2.2.18 S Command - Save tracking parameters

    The **S** command directs subroutine SAVE to write all data in COMMON blocks to an external file. It allows the user to suspend tracking at any point in the program. Once all the tracking data is SAVE'd, it may be restored simply by issuing the **G** command (see next section). Subroutine SAVE is also useful for those who might want to analyze ESME tracking data independently of the program.[6] The format of the **S** command is as follows:

```
S
filename
```

where *filename* is the name of the external file to which the data is to be written.

2.2.19 G Command - Get tracking parameters

    The **G** command directs subroutine GET to read data from an external file into ESME's common block data. It is intended to be used to retrieve data written using the **S** command. The **G** command format is very similar to that for the **S** command:

```
G
filename
```

where *filename* is the name of the external file from which the data is to be read. Because **G** recovers all parameters, input data for a new run could consist of only **G**, **T**, and **Q** commands for example. It is useful for restarting a long calculation from a checkpoint or for making alternative conclusions to a common intermediate result.

    The contents of the saved file are specific to a particular version of the program. A **G** command is likely provide spurious data if it is accessing a save file written by an earlier program version.

---

[6] See, for example, Appendix B.

Chapter 3

Using The Program

3.1   Running the Program

Although the graphics can display to the screen during execution, ESME is not usually run as an interactive program. It reads data from standard input and writes numerical results and error messages to standard output. Errors detected by the system or arithmetical library go to the standard error file. However, with command line switches input may be taken from an alternate file and pauses set after each display. Additional output files are used with names containing the fortran unit number as an extension. The names for auxiliary input files are read in with the data for the commands that call for the auxiliary input. The fortran units used are

> unit 07 – input to restore previous running condition; input to GET
> unit 08 – SAVE output for check pointing or interrupted calculation
> unit 09 – history data (output)
> unit 10 – RF voltage, frequency, and phase curves
> unit 11 – longitudinal impedance table
> unit 12 – table of resonances for longitudinal impedance calculation
> unit 14 – graphical output in postscript — esme.ps
> unit 15 – bunch length table for voltage feedback control
> unit 16 – basis function for time domain collective voltage calculation
> unit 18 – output for graphics post-processer
> unit 19 – tabular output from HISTORY (user plots and analysis)
> unit 20 – mountain range plot output data

ESME can be run directly from a command line under UNIX-like systems:

> esme2000 [optional switches] $< input\ 1 > output\ 2 > errout$

The available command line switches and their functions are

```
USAGE:   esme [args [parameters]]
-----
where [args] is one or more of the following

-i            : interactive output (pauses between plots)
-a4           : output on a4 paper
-bw           : turn color output off
-d int        : debugging message level
-f filename   : use filename for input instead of standard input
-h            : help
-o filename   : use filename for output instead of standard output
-p filename   : use filename for postscript graphics instead of esme.ps
-V            : version number
```

If the -f switch is used to designate the input data file, files generated by the program will be named with whatever precedes a
"." (if any) in the input filename. Oherwise the produced filenames will be "UNDEFINED.un", where "un" is the fortran unit
number.

It is possible to run ESME interactively with the commands and namelists entered by the user directly from the keyboard. One
can also specify input and output files from the command line and request pauses between plots to the screen, called interactive
output mode.

If the post-processor option[1] is in effect, then ESME writes its data to the file associated with FORTRAN logical unit 18
(every MPLOT turns, or upon issuance of the **D** command). History records are written to the file associated with unit logical
unit 9. These files may be processed later according to methods of the user's own choosing. A simple post-processor has been
written which processes these files using routines from ESME. It is described in Appendix B.

3.2   Input Structure

In this section the format required of various input files is described. First and foremost is FORTRAN logical unit 5 (*i.e.*
standard input), which is the main input file for ESME and often the only input file. In addition, ESME may read tables of values
for the RF voltage, frequency, or phase as functions of time, the magnet ramp, the wall impedance as a function of frequency, a
table of resonant impedances, a table of nominal bunch width *vs.* time for controlling feedback to rf amplitude, and a tabulation
of a basis function for time domain calculations. These auxiliary files are read fully, so the tables can cover any time during the
calculation. In each case the name of the auxiliary file (complete path) is passed via a namelist character variable FILxxx. New
tables can be introduced by naming a new file.

3.2.1   Command file

The commands which ESME accepts have already been described in Chapter Two. Each command is indicated by a single
uppercase letter or a numeral in the first column of a line. The following four places in the line are discarded, and the rest of the
line is read[2] and echoed in the output along with the command as a comment. Following many of the commands is namelist data.
The namelist input is specified by entering a dollar sign (or &) in the *second* column of the input file, followed immediately
by the namelist identifier, after which any variable which is a member of the namelist may be entered by specifying its name
followed by an equal sign followed by the value to be assigned. The assignment of logical variables will depend on the particular
machine (e.g. .TRUE., .T., or T). The namelist is terminated by a $END (or simply $ or /). All of the command examples given
previously are valid f77 namelist entries.

3.2.2   RF curves

Each of the NRF rf systems can have a voltage program and a phase or frequency program. If both frequency and phase
programs are provided, the phase program will take precedence. There are options for simple programs provided with the **A**
command (see 2.2.2), but if these are not sufficient, the desired curves can be provided in tables read from an external file. The
curves file will be read when FILCRV is given. If other curve tables will be used in a later **A** command, they can be included in
the same file. This subsection describes the content and format of the curves file.

The numerical tables representing the curves are identified by character string headers which are explained later. All in-
put records are free-format (list-directed). The first numerical record for each table is an integer giving the number of records
to follow. The table entries follow with a time and a curve value per record. The standard compilation limits the individual
curves to 350 or fewer points. Just as for the simple curve options available from the program, the initial and final values of
the programmed quantities are set by the **A** command parameters VI(i), VF(i), PSII(i), PSIF(i), FRI(i), FRF(i). This means it
is not necessary to re-compute a complete curve to effect a change in one or both end points. All curves, ramp or rf, internally
generated or provided in a table, are normalized in the same way:

$$f_i(t_i) = [T(t_i) - T(t_\circ)]/[T(t_f) - T(t_\circ)]   ,$$

where the T's are table values, t_i are the tabulated times, and $t_\circ$ and t_f are the initial and final times from the table. The contin-
uous function $f(t)$ is obtained by spline interpolation from the normalized table $f_i(t_i)$.

For example, to make voltage values used by the program identical to those appearing in the table, set VI equal to the program
value for t=TVBEG and VF to the program value for t=TVEND. The normalization is generally convenient, but one must avoid
putting in curves with the same value for the initial and final entries. Such functions can be handled by extending the table on

---

[1]POSTP=T in the **O** command.
[2]FORTRAN format (A1,4X,A74).

one end or the other beyond the time range being used. There is no rigid relationship between the table times and the begin and end times in the **A** command data, but $t_\circ$ should not be later than TVBEG and t_f should not be earlier than TVEND.

The spline scheme assumes zero derivatives outside the range of the table when interpolating in the end intervals. Usually the assumption is satisfactory, but it is conservative to extend the tables a couple of intervals at each end to ensure the correct derivatives over the full time range needed.

When the program opens the curves file it looks for a record with a character string "SOURce i", where "i" is a required integer identifying the rf system to which the following table(s) apply. All character headers are checked for the first four characters only; these characters must be all caps. Thus, SOUR, SOURCE, SOURce are fine but source or Source will not be recognized. These source headers can be in any numerical order, and they can be repeated for entering separate voltage, frequency, and phase programs for the same system if desired. It is also acceptable to read in more than one curve under a single source heading, but when this is done the tables must appear in the order voltage first followed by frequency, if present, and finally phase, if present. The individual tables are identified by the character headers VOLTage, FREQuency, and PHASe, where again only the first four characters are significant. Schematically a curves file might look something like the following:

```
SOURce 1
VOLTS
 5
0.     1.0
 .001  1.12
 .002  1.36
 .003  1.84
 .006  2.00
FREQ
  25
0.     37.1101
 .001  37.1175
 .002  37.1203
    .
    .
    .
SOUR 2
VOLTAGE
 5
.01  2.00
.011 1.8
.013 1.7
.014 1.1
.016 1.0
SOURCE 2
PHAS
       150
   .
   .
   .
```

etc.

3.2.3 Magnet ramp table

If the KURVEB=4 option is taken for the **R** command, one is to provide the name (complete path) of a file containing the ramp function by setting the character variable FILRMP in the /RING/ namelist. The file is opened and assigned to FORTRAN unit 13. The first line of the table in list directed format is an integer, the number of table lines to follow. Each succeeding line contains a time value and a ramp value. The scaling of the ramp values is arbitrary; the ramp is turned into a normalized spline table. The scaling is determined by the /RING/ variables W0I or P0I and W0F or P0F. The standard compilation limits the table to 350 or fewer entries. The first time entry should be at or before TI, and the final entry should be at or after TF. Having values outside the TI – TF range may provide a smoother $\dot{B}$ by properly setting the slopes at the end points — so-called guard values.

3.2.4 Impedance table

The **B** command allows the user to enter a table for the wall impedance as a function of frequency. The file name (complete path) is given as the value of the character variable FILIMP read by the /SCHG/ namelist. The file will be opened, assigned to FORTRAN unit 11.The first record is an integer giving the number of table entries to follow. Each following record consists of frequency (in MHz) accompanied by a real and imaginary part. The format of the input is as follows:

NZ f^1 x^1 y^1
.
.
.
f^{NZ} x^{NZ} y^{NZ}

The frequency is in MHz; the impedance $Z^n = x^n + iy^n$ is in Ohms. The standard compilation limits the impedance table to 350 entries or less. The FFT used requires a negative sign for a capacitive reactance, *i. e.*, $Z = R - i(X_C - X_L)$. The impedance should represent the value for the entire circumference, not the value per unit length. The FORTRAN format for each line is list-directed, so it is sufficient to list the values separated by commas or spaces in the file. The entries should be made in order of increasing frequency ($f^1 < f^2 < \ldots < f^{NZ}$). Impedance values will be calculated by cubic spline interpolation from the table. As for the rf tables, it may give a better curve if the function is continued two tabular intervals below and above the frequencies that will be needed for the problem. These guard values will ensure that the derivatives in the end intervals are correctly represented and avoid possible wild interpolation near the ends of the curve.

3.2.5 Resonance table

Resonant impedances may be specified in a simple form. The resonance values are read from the file whose name is specified by the FILRES variable in the /SCHG/ namelist.[3] This file is assigned to input unit 12. The first record is an integer giving the number of resonances to follow. Each resonance is given in a single record. The format is list directed, where the values to be read in are:

frequency [MHz],   real shunt impedance at resonance [Ω],   multiplier,   Q

where the multiplier, which is usually 1., multiplies the impedance which is calculated from the frequency, shunt impedance, and Q values for the resonance. However note that, as mentioned in Section 2.2.12, a multiplier of zero has a special purpose. When the time domain calculation of resonator response has been activated by setting QREZON true, those resonances with a zero multiplier will be treated with the time domain technique.

3.2.6 Time domain basis table

If the FILTDB string in the **B** command is not 'DUMMY', a basis function or characteristic solution for the time domain response of some longitudinal impedance is read from the file specified by FILTDB. The time interval is the first record and one number from each following record is the value of the basis function for each successive time step. The appropriate length for the table is either the length of time for which the function has a significant amplitude or the circulation period divided by the periodicity, i. e., τ/FRAC. Taking FRAC > 1 corresponds to the physical situation where the impedance occurs multiple times in the ring. The basis function is the time response of the impedance to a triangular pulse of unit charge and length $2 \times \Delta$, where Δ is the tabular interval. The interval should be short enough that a binning of the charge distribution at the same interval gives an adequate representation. Table size is limited by the compiled value of MAXCVB, 1000 in the standard compilation. However, the **Y** command can be used to allocate a larger table space with KMAXCVB (see section 2.2.17).

3.2.7 Bunch width table

One can damp quadrupole mode bunch oscillation (*i.e.* shape oscillation) by feeding back from bunch length measurement to the rf voltage. However, bunch length usually has a secular variation arising from voltage change or change in η with energy. Therefore, one needs to specify a transfer function for the feedback which does not confuse the desired secular change with the undesired oscillation. Since this transfer function will be problem dependent, it can not be built into the code. However, when one specifies the parameter time variation for a problem one knows enough to calculate the corresponding nominal bunch length. With a table of the expected bunch length *vs.* time, the feedback can be applied to just the error between actual and nominal. The actual length need not be correct so long as the functional form is right because the program normalizes the curve to the initial

---

[3] see Section 2.2.12

length. Thus, one table will usually work quite well over a substantial range of initial emmitance. The times and bunch lengths are read from the file whose name is specified by the FILBWD parameter in the /LLRF/ namelist.[4] This file is assigned to input unit 15. The first line is an integer giving the number of lines to follow. Each succeeding line contains a time [s] and a bunch length in any convenient units separated by a comma or space in the usual list-directed convention. Cubic spline interpolation is used between table values, so it is not likely that many entries will be required. Like the rf curve tables, the bunch length table is limited to a maximum of 350 entries.

3.2.8 Examples

Below are some complete data sets for reasonably simple and conventional cases. Between the included comments, the information in section 2.2, and the output produced by running with the sample data, it should be possible to understand and use most program features. If one is trying to use the program to study multiparticle dynamics issues, it may be necessary to do some trial runs to understand what the program will do; ESME is not an expert system for collective effects, but it does provide powerful tools for a careful user.

```
=================== Bunch Pair Merging in CERN PS ============================
W    Test of adiabatic bunch pair merging in the PS
Y    This memory allocation reduces the default allocation.
 $MEMORY  KNPHASE=2001 $
R    CERN PS at 3.57 GeV/c
 $RING REQ=100., GAMTSQ=37.21, W0I=2753., FRAC=5. $
A    h=20 and h=10 RF systems: turn down h=10 and turn up h=20 linearly.
 $RF NRF=2 H=20,10 VI=40.E-3,4.E-3 VF=4.E-3,40.E-3 TVEND=0.020,0.040
  KURVE=1,1 PSII=0.,-90. $
P    Parabolic bunch 1; center it in h=20 bucket. Distinguish it from bunch 2.
 $POPL8 KIND=13 SBNCH=0.4 NPOINT=1000 THOFF=-9. PARTION=.T. $
P    Parabolic bunch 2. Center in next h=20 bucket.
 $POPL8 THOFF=9. $
A    RF for tracking. Only the phases have changed; everything else is the same.
 $RF  PSII=180.,0. $
O    Output format: Plot Delta E, Delta Theta scatter plot every 5000 turns.
 $GRAPH MPLOT=5000 IDEV=1 IMETA=1 PLTSW(8)=.F. PLTSW(10)=.F.
  IRF=0 THPMIN=-18. THPMAX=18. DEPMIN=-15. DEPMAX=15.
  TITL='BUNCH PAIR MERGING IN THE PS' $
D    Plot at start.
M    Set up Mountain Range for a trace every 200 turns.
 $MRANGE MRMPLOT=200 IPU=T $
T    Tracking conditions: just go for 0.035 s.
 $CYCLE TTRACK=0.035 $
N    Plot mountain range with two iterations of local smoothing
 $MRPLOT SMOOTH=-2 NTRACE=80 SCALE=0.2 MRPMIN=-18. MRPMAX=18.  $
Q    ESME stop.


=================== Basic Data for Crossing Transition =======================
W    Fermilab MR: Normal transition crossing SIMPLIFIED (but still realistic)
W    pdot @trans about  83 GeV/c/s
R    MR: Start part way into cycle.
 $RING REQ=1000  p0i=9750. p0idot=25.83E3 p0f=26900 p0fdot=120.0E3
  KURVEB=6 GAMTSQ=355.32 ALPHA1=.0023237 FRAC=1113 TI=0.23035 TF=.46556
  TSTART=0.23035  $
A    Let program calculate voltage for constant area bucket
 $RF H=1113 ISYNC=1 vkon=f holdba=t VI=2. $
O    Plot bucket and rf waveform every 3000 turns.
 $GRAPH DEPMIN=-200. DEPMAX=200. MPLOT=3000 IDEV=1 IMETA=1
  THPMIN=-0.097 THPMAX=0.097 PLTSW(8)=F PLTSW(10)=F PLTSW(5)=T
```

---

[4]See subsection 2.2.11

```
      TITL=''MR(normal tr. x'ing) Constant area Bucket''  $
P     0.1 eVs parabolic bunch
 $POPL8 KIND=14 SBNCH=.1 NPOINT=200  $
D     Plot initial distribution
T
 $CYCLE TSTOP=0.46556 HISTRY=T $
D     Plot at very end
R     M R Ramp II: Linear portion of ramp
 $RING JNRAMP=T TF=.79139 p0f=66000. KURVEB=1 $
T
 $CYCLE TSTOP=0.645  $
D
H
 $HISTRY
  NPLT=1,2,1,3,1,4,1,5,1,6,1,7,1,13,1,14,1,15,1,101,1,32,1,17,1,17  $
Q
==================== ''Realistic'' Transition Crossing =========================
W     Fermilab MR :  Normal transition crossing
W     N_q=2.5e10 Z_||/n=8.6 PIPRAD = 2.54cm
W     This example involves a couple of external files.
W     pdot @trans about  83 GeV/c/s
R     MR: Start part way into cycle; discard particles > 2.54 cm off central orbit
 $RING REQ=1000  p0i=9750. p0idot=25.83E3 p0f=26900 p0fdot=120.0E3
  KURVEB=6, GAMTSQ=355.32, ALPHA1=.0023237 FRAC=1113 TI=0.23035 TF=.46556
  TSTART=0.23035 PIPRAD=0.0254 EBDRY=T  $
A     Use a voltage curve from an external file
 $RF H=1113 ISYNC=1 VI=2. VF=3.699412 TVBEG=.23035 TVEND=.6455522 KURVE=4
  FILCRV='mrtrramp.crv'  $
O
 $GRAPH DEPMIN=-200. DEPMAX=200. MPLOT=3000 IDEV=1 IMETA=1
  THPMIN=-0.097 THPMAX=0.097 NBINTH=100 DELCON=0.005 NPJMP=10
  PLTSW(8)=F PLTSW(10)=F PLTSW(5)=T
  TITL=''MR(normal tr. x'ing): N_q=2.5e10, Z_||/n=8.6''  $
P  Going to look at collective volts up to 3 GHz; need AT LEAST this many.
 $POPL8 KIND=14 SBNCH=.1 IPOP=1 NPOINT=20000  $
B     Q=1 resonance (described on external file) to represent broadband Z_||/n
 $SCHG SCON=T FDON=T ENQ=2.5e10 A=4.E-3 B=2.5E-2
  NBINSC=128 TCHGON=0.0 NBINFFT= 128, NIXNOIS=1 FILIMP='mrzon_8pt6ohm.dat'  $
L     Use phase feedback to reduce dipole oscillations.
 $LLRF PHFBON=T  $
D
T
 $CYCLE TSTOP=0.46556 HISTRY=T $
D
R     M R Ramp II: Linear portion of ramp
 $RING JNRAMP=T , p0fdot=120.0E3 p0idot=120.0E3
  TF= .79139 p0f= 66000. TI = 0.46556 KURVEB=1 $
D
T
 $CYCLE TSTOP=0.645  $
D
H
 $HISTRY
  NPLT=1,2,1,3,1,4,1,5,1,6,1,7,1,13,1,14,1,15,1,101,1,32,1,17,1,17  $
Q
--------------------------------------------------------------------------------
 1
```

```
1.7000135E+3 3.06203E5 1 1
FOR MAIN RING WITH Z/N=8.6 OHMS AND CUT OFF OF 1.7 GHZ (File mrzon_8pt6ohm.dat)
--------------------------------------------------------------------------------
SOURCE    1
 VOLTS
        138                                                        (File mrtramp.crv)
  2.303500E-01  2.000000E+00 \ 2.353401E-01  2.033759E+00 \ 2.403302E-01  2.062870E+00
  2.453406E-01  2.088300E+00 \ 2.503482E-01  2.109665E+00 \ 2.553348E-01  2.127636E+00
  2.603424E-01  2.142347E+00 \ 2.653290E-01  2.154087E+00 \ 2.703366E-01  2.163358E+00
  2.753443E-01  2.170156E+00 \ 2.803308E-01  2.175268E+00 \ 2.853368E-01  2.178607E+00
  2.903373E-01  2.180647E+00 \ 2.953378E-01  2.181779E+00 \ 3.003384E-01  2.181987E+00
  3.053389E-01  2.181473E+00 \ 3.103394E-01  2.180420E+00 \ 3.153400E-01  2.178910E+00
  3.203405E-01  2.176697E+00 \ 3.253410E-01  2.173683E+00 \ 3.303415E-01  2.169721E+00
  3.353421E-01  2.164204E+00 \ 3.403426E-01  2.156876E+00 \ 3.453431E-01  2.146863E+00
  3.503435E-01  2.133029E+00 \ 3.553370E-01  2.114116E+00 \ 3.603304E-01  2.087458E+00
  3.608339E-01  2.084188E+00 \ 3.613375E-01  2.080732E+00 \ 3.618410E-01  2.077240E+00
  3.623446E-01  2.073459E+00 \ 3.628481E-01  2.069704E+00 \ 3.633307E-01  2.065883E+00
  3.638342E-01  2.061674E+00 \ 3.643377E-01  2.057324E+00 \ 3.648413E-01  2.052737E+00
  3.653448E-01  2.047943E+00 \ 3.658274E-01  2.043135E+00 \ 3.663309E-01  2.037896E+00
  3.668345E-01  2.032339E+00 \ 3.673380E-01  2.026512E+00 \ 3.678415E-01  2.020364E+00
  3.683451E-01  2.013773E+00 \ 3.688276E-01  2.007175E+00 \ 3.693312E-01  1.999734E+00
  3.698347E-01  1.991851E+00 \ 3.703383E-01  1.983333E+00 \ 3.708418E-01  1.973973E+00
  3.713453E-01  1.963920E+00 \ 3.718279E-01  1.953103E+00 \ 3.723314E-01  1.940498E+00
  3.728350E-01  1.926325E+00 \ 3.733385E-01  1.909210E+00 \ 3.738421E-01  1.888088E+00
  3.743456E-01  1.858132E+00 \ 3.748282E-01  1.80        \ 3.753400E-01  1.910000E+00
  3.758226E-01  1.961538E+00 \ 3.763261E-01  2.004583E+00 \ 3.768297E-01  2.041282E+00
  3.773332E-01  2.073507E+00 \ 3.778367E-01  2.102951E+00 \ 3.783193E-01  2.129045E+00
  3.788228E-01  2.154578E+00 \ 3.793264E-01  2.178674E+00 \ 3.798299E-01  2.201581E+00
  3.803335E-01  2.223394E+00 \ 3.808370E-01  2.244314E+00 \ 3.813196E-01  2.263748E+00
  3.818231E-01  2.283351E+00 \ 3.823266E-01  2.302127E+00 \ 3.828302E-01  2.320337E+00
  3.833337E-01  2.338108E+00 \ 3.838373E-01  2.355425E+00 \ 3.843198E-01  2.371690E+00
  3.848234E-01  2.388211E+00 \ 3.853269E-01  2.404248E+00 \ 3.858304E-01  2.420084E+00
  3.863340E-01  2.435565E+00 \ 3.868375E-01  2.450683E+00 \ 3.873201E-01  2.465123E+00
  3.878236E-01  2.479796E+00 \ 3.883272E-01  2.494084E+00 \ 3.888307E-01  2.508291E+00
  3.893342E-01  2.522257E+00 \ 3.898378E-01  2.535978E+00 \ 3.903203E-01  2.548979E+00
  3.953348E-01  2.674677E+00 \ 4.003282E-01  2.787153E+00 \ 4.053216E-01  2.890317E+00
  4.103360E-01  2.986864E+00 \ 4.153295E-01  3.077085E+00 \ 4.203229E-01  3.162423E+00
  4.253373E-01  3.243913E+00 \ 4.303308E-01  3.321436E+00 \ 4.353242E-01  3.395712E+00
  4.403386E-01  3.467440E+00 \ 4.453321E-01  3.536502E+00 \ 4.503255E-01  3.603278E+00
  4.553399E-01  3.668189E+00 \ 4.603334E-01  3.730967E+00 \ 4.655576E-01  3.794616E+00
  4.705510E-01  3.800847E+00 \ 4.755445E-01  3.805519E+00 \ 4.805379E-01  3.808780E+00
  4.855523E-01  3.811178E+00 \ 4.905458E-01  3.812485E+00 \ 4.955392E-01  3.813066E+00
  5.005536E-01  3.812824E+00 \ 5.055470E-01  3.812006E+00 \ 5.105405E-01  3.810700E+00
  5.155549E-01  3.808847E+00 \ 5.205483E-01  3.806463E+00 \ 5.255418E-01  3.803827E+00
  5.305562E-01  3.800888E+00 \ 5.355496E-01  3.797649E+00 \ 5.405431E-01  3.794283E+00
  5.455575E-01  3.790346E+00 \ 5.505509E-01  3.786541E+00 \ 5.555444E-01  3.782531E+00
  5.605378E-01  3.778366E+00 \ 5.655522E-01  3.774137E+00 \ 5.705457E-01  3.769613E+00
  5.755391E-01  3.765215E+00 \ 5.805535E-01  3.760652E+00 \ 5.855470E-01  3.756090E+00
  5.905404E-01  3.751374E+00 \ 5.955548E-01  3.746831E+00 \ 6.005483E-01  3.742003E+00
  6.055417E-01  3.737262E+00 \ 6.105561E-01  3.732544E+00 \ 6.155496E-01  3.727885E+00
  6.205430E-01  3.723113E+00 \ 6.255574E-01  3.718336E+00 \ 6.305509E-01  3.713484E+00
  6.355443E-01  3.708887E+00 \ 6.405377E-01  3.704164E+00 \ 6.455522E-01  3.699412E+00
================================================================================
(Note: If you want to scan the above voltage table to try this example, you will
need to replace the \'s with carriage returns to get the table format of one time
and one voltage per record.)
```

Chapter 4

Customizing the Code

In this chapter some of the basic structure of the program is described, and the use of pre-compilation directives in adapting the code is discussed. The material in this section is not ordinarily needed to use the program. This section alone is not likely to make the reader an ESME expert; however, the material presented here will be useful and perhaps even sufficient for someone wanting to add a custom feature via the SHAZAM entries. Some of this information should also be useful to those needing to install the code in a new environment.

4.1 Program Basics

The first part of this section describes the overall structure of ESME. The second section details the main tracking loop. The third section lists the important variables in ESME, which are held in COMMON and thus available in SHAZAM.

4.1.1 Program structure

The basic pattern of ESME is a main program which calls subroutines selected by single letter commands in the input stream. The called routine reads in any needed parameters; it and any dependent subroutines carry out calculations for a distinct phase of the calculation or for a distinct accelerator subsystem. The program is integrated by putting particle coordinates and system variables into named commons each of which contain a group of closely related quantities. Higher level subroutines communicate through common. Certain lower level routines and some utility routines shared among different functional areas pass data through calling lists. Nearly all system and coordinate variables are stored in common blocks; only loop counters and a few intermediate results are local variables. Thus, the program is divided into numerous functional modules but important variables are global. The large number of parameters often required to specify a distribution and the rf systems which act upon it encourage this structure. The structure has also proved amenable to extension without major interference with existing features.

4.1.2 Main tracking loop

The fundamental mapping or tracking algorithm is contained in the subroutine CYCPROG which is called when a **T** command is encountered in the data stream. The subroutine reads data for the number of turns to track *etc.* and then tracks successive turns applying the difference equations to each particle on each turn. At the end of each turn the tracking duration is checked, various properties of the distribution are calculated, system parameters are updated, and tests are applied to see if any selected parameter has reached a desired endpoint. Optional calls to one of the 10 SHAZAM entries can be made either preceding or following an iteration. For most purposes the alternatives are equally satisfactory, but some quantities are re-initialized at the beginning of turns and can not be changed by a SHAZAM call at the end.

4.1.3 Important variables

The various named common blocks are listed below. Each entry includes a statement of purpose and a list of all the included variables. If many of the variables are input data, the command that reads them is noted. Because input variables are defined in the command tables, they are not separately identified in the following listing. However, other variables in common are described in the comment lines that begin with an exclamation point (!). In the common /SPCHG/ there are some work arrays that are not described in either place; the intention is to discourage any use.

```
C
C  bktsup.inc (edited)
C  Block parameterizing isolated or barrier bucket generation.
C
      DOUBLE PRECISION THL,THU
! Lower and upper phase bounds for truncated rf waveform
C
      COMMON /BKTSUP/ THL(NSRC),THU(NSRC)
C  ----------------------------------------------------------------------
C
C  blankcom.inc (edited)
C  THE MAIN STORAGE BLOCK. CONTAINS NPHASE PHASE SPACE COORDINATES AND
C  THE POINTERS TO SEPARATE PARTITIONS OF THE DISTRIBUTION
C
      LOGICAL PARTION
C
#ifdef DYNMEM
C
#ifndef _DYNMEM_INC
#include "dynmem.inc"
#endif
C  If dynamic memory allocation is enabled, PHASETH and PHASEE have length of
C  KNPHASE read by Y command
      COMMON /BLANK/   DUMMY1,DUMMY2
      COMMON /IBLANK/  KOUNT,KNTSC,KLASSES,KLIMIT(0:NKLIM)
      COMMON /LBLANK/  PARTION
C
#else
C
      DOUBLE PRECISION PHASETH, PHASEE
      COMMON /BLANK/  PHASETH(0:NPHASE),PHASEE(0:NPHASE)
! The phase and energy coordinates of the macroparticles
      COMMON /IBLANK/ KOUNT,KNTSC,KLASSES,KLIMIT(0:NKLIM)
! KOUNT number of macroparticles, KNTSC number which contribute to beam charge,
! KLASSES number of partitions of macroparticle distribution, KLIMIT index of
! partition boundaries, i.e., first particles in each class
      COMMON /LBLANK/ PARTION
C
#endif
C  ----------------------------------------------------------------------
C
C  bucket.inc (edited)
C  BUCKET PARAMETERS TURN-BY-TURN
      DOUBLE PRECISION GNUS,SBCKT,HBCKT
      COMMON /BUCKET/ GNUS,SBCKT,HBCKT
! GNUS synchrotron tune, SBCKT bucket area, HBCKT bucket height
C  ----------------------------------------------------------------------
C
C  bunch.inc (edited)
C  BUNCH PARAMETERS TURN-BY-TURN
      DOUBLE PRECISION THBAR,EBAR,THRMS,ERMS,EPSILON,SBUNCH,ANORM
      COMMON /BUNCH/ THBAR,EBAR,THRMS,ERMS,EPSILON,SBUNCH,ANORM
! THBAR average phase, EBAR average energy, THRMS rms phase spread,
! ERMS rms energy spread, EPSILON rms emittance (unless modified by
! ANORM .NE. 1), ANORM renormalizing factor
C  ----------------------------------------------------------------------
C
```

```
C  const.inc (edited)
C  CONTAINS MATHEMATICAL AND PHYSICAL CONSTANTS, PROGRAM CONSTANTS, AND
C  CONVERSION FACTORS FROM EXTERNAL TO INTERNAL UNITS
      DOUBLE PRECISION HALFPIE,PIE,TWOPIE,FOURPIE,RADDEG,DEGRAD,
     1     C,EMPCSQ,EMECSQ,RE,RP,QE,ZNAUGHT
      COMMON /CONST/ HALFPIE,PIE,TWOPIE,FOURPIE,RADDEG,DEGRAD,
     1     C,EMPCSQ,EMECSQ,RE,RP,QE,ZNAUGHT
! Multiples of pi, angle conversions, velocity of light, proton and
! electron rest energies, electron and proton classical radii, free space
! impedance
C  ----------------------------------------------------------------------
C
C  current.inc (edited)
C  MISCELLANEOUS PARAMETERS AND INTERMEDIATE RESULTS TURN-BY-TURN
C
      DOUBLE PRECISION ES,PS,RS,BETAS,BETASQ,ETA,E0,P0,BETA0,
     1     THREF,EREF,TAU,TIME,DEBFLD,PDOT,DP0DT,ELO,EHI,TGCURR,GMSQINV,
     2     DELR,PHIS,DELES,EV,PSI0,PSI,FREQ,SBINIT,HBINIT,RSCALE
      COMMON /CURRENT/ ES,PS,RS,BETAS,BETASQ,ETA,E0,P0,BETA0,
     1     THREF,EREF,TAU,TIME,DEBFLD,PDOT,DP0DT,ELO,EHI,TGCURR,GMSQINV,
     2     DELR,PHIS,DELES,SBINIT,HBINIT,RSCALE,
     3     EV(NSRC),PSI0(NSRC),PSI(NSRC),FREQ(NSRC)
! ES synchronous energy, PS synchronous momentum, RS synchronous radius,
! BETAS synchronous velocity, BETASQ beta_s^2, ETA time slip factor,
! E0 central orbit energy, P0 central momentum, BETA0 central velocity,
! RSCALE factor for stepping up mapping interval, THREF,EREF theta-E
! coordinates of reference particle, TAU beam circulation period, TIME
! calculated time, DEBFLD energy change per turn of central orbit, PDOT
! rate of synchronous momentum change, DP0DT rate of change of central
! momentum, ELO,EHI limits of energy corresponding to PIPRAD input parameter,
! TGCURR time of starting gamma-t jump, GMSQINV gamma^-2, DELR=RS-REQ,
! PHIS synchronous phase at harmonic number of 1, DELES synchronous energy
! change per turn, SBINIT & HBINIT initial values of bucket area and height
! at start of T command, RSCALE map scaling parameter, EV the rf voltages,
! PSI0 the program value of phases in ! radians, PSI adjusted phases,
! FREQ rf frequencies
      COMMON /ICURRENT/ M,MM,NTURNS,MHIST,MOST
! M turn number in current T command, MM turn number, NTURNS turns counter
! for writing history record, MHIST calculated turns interval for writing
! history records, MOST the index for the principal rf system (greatest
bucket height)
      COMMON /LCURRENT/ TRPSW
! flag for traps in CYCPROG active
      LOGICAL TRPSW
C  ----------------------------------------------------------------------
C
C  curves.inc  (edited)
C  PHASE SPACE COORDINATES OF BUCKET OR OTHER CONTOUR OF INTEREST
      DOUBLE PRECISION CURVE,TURNPT,TOP,BOTTOM
C
      COMMON /CURVES/ CURVE(ICURVE,2),TURNPT(4,3),
     1     TOP(2,ICURVE/2),BOTTOM(2,ICURVE/2),NC
! CURVE theta,E coordinates of a curve to be plotted on phase plane,
! TURNPT the turning points of the curve (third coordinate is number of the
! point along the curve), TOP, BOTTOM coordinates for top and bottom halves
! of closed curve, NC number of points on curve
C  ----------------------------------------------------------------------
```

```fortran
C
C  cyclp.inc  (edited) Parameters not described are from the T command.
C  PARAMETERS GOVERNING DURATION & OPTIONAL FEATURES OF THE TRACKING CALCULATION
C
      DOUBLE PRECISION TBEGIN,TEND,TSTART,TSTOP,TTRACK,
     1      RSCALE0,THREF0,EREF0,ETATRP,PHISTRP,HISTSIZ
       COMMON /CYCLP/ TBEGIN,TEND,TSTART,TSTOP,TTRACK,
     1      RSCALE0,THREF0,EREF0,ETATRP,PHISTRP,HISTSIZ
       COMMON /ICYCLP/ MSTEP,LGRTHM,ITRAP(NTRP),MGRACE(NTRP),NTRAP
! NTRP is a compiled parameter (parameters.inc)
       COMMON /LCYCLP/ HISTRY,BBDRY,EAPROX
       LOGICAL HISTRY,BBDRY,EAPROX
C  ----------------------------------------------------------------------
C
C  dynmem.inc (edited)
#ifndef _DYNMEM_INC
#define _DYNMEM_INC
       DOUBLE PRECISION
     1          PHASETH(0:NPHASE), PHASEE(0:NPHASE)
       POINTER (PTRPHASETH,PHASETH)
       POINTER (PTRPHASEE,PHASEE)
C
       REAL SPHASETH(NPHASE)
       REAL SPHASEE(NPHASE)
       POINTER (PTRSPHASETH,SPHASETH)
       POINTER (PTRSPHASEE, SPHASEE)
C
       DOUBLE PRECISION EVRES(MAXCVB),ZRES(MAXCVB)
       POINTER (PTREVRES,EVRES)
       POINTER (PTRZRES,ZRES)
C
       DOUBLE PRECISION DQ(MAXCVB),THTA(MAXCVB),DQ2(MAXCVB)
       POINTER (PTRDQ,DQ)
       POINTER (PTRTHTA,THTA)
       POINTER (PTRDQ2,DQ2)
C
       DOUBLE PRECISION DQDT(MAXCVB)
       POINTER (PTRDQDT,DQDT)
C
       DOUBLE PRECISION USC(MAXCVB)
       POINTER (PTRUSC,USC)
C
       DOUBLE PRECISION EVTD(MAXCVB),FIT(MAXCVB),TDTBL(MAXCVB)
       POINTER (PTREVTD,EVTD)
       POINTER (PTRFIT,FIT)
       POINTER (PTRTDTBL,TDTBL)
C
       REAL BIN1(MAXCVB), BIN2(MAXCVB), XDHIS(MAXCVB)
       POINTER (PTRBIN1,BIN1)
       POINTER (PTRBIN2,BIN2)
       POINTER (PTRXDHIS,XDHIS)
C
       DOUBLE PRECISION BLDQ(MAXCVB),ZBL(MAXCVB),EVBL(MAXCVB)
       POINTER (PTRBLDQ,BLDQ)
       POINTER (PTRZBL,ZBL)
       POINTER (PTREVBL,EVBL)
C
```

```
       DOUBLE PRECISION BLI(MAXCVB)
       POINTER (PTRBLI,BLI)
C
       REAL PLTX(MAXPTS),PLTY(MAXPTS)
       POINTER (PTRPLTX,PLTX)
       POINTER (PTRPLTY,PLTY)
C
       COMMON /DYNPTR/ PTRSPHASETH,PTRSPHASEE,
      1                PTREVRES,PTRZRES,
      1                PTRDQ,PTRTHTA,PTRDQ2,PTRDQDT,
      1                PTREVTD,PTRFIT,PTRTDTBL,
      1                PTRBIN1,PTRBIN2,PTRXDHIS,
      1                PTRBLDQ,PTRZBL,PTREVBL,PTRBLI,
      1                PTRPLTX,PTRPLTY
#endif
C
C  -----------------------------------------------------------------
C
C  feeds.inc  (edited) Parameters not described are from the L command.
C  PARAMETERS DEFINING FEEDBACK LOOPS (CURRENTLY PHASE & VOLTAGE - NO RADIAL)
C
       DOUBLE PRECISION DLIMIT,PLIMIT,FBFACT,VLIMIT,VFBFCTR,PSIADD,DAMPL,
      1       FBPRS,W,BLTABL1,BLTABL2,BLTABL3
       COMMON /FEEDS/ DLIMIT,PLIMIT,FBFACT,VLIMIT,
      1       VFBFCTR,PSIADD,DAMPL,FBPRS(IFBPRS),W(IFBPRS),
      2       BLTABL1(LTABL),BLTABL2(LTABL),BLTABL3(LTABL)
! PSIADD phase feedback adjustment to phis, DAMPL voltage feedback
! adjustment factor, FBPRS circular buffer for bunch centroid phases
! (IFBPRS from parameters.inc), BLTABL spline table of bunch width vs time
! for voltage feedback
       COMMON /IFEEDS/ NTUAVG,NTURES,ITFB,NTL
! NTL length of bunch length table
       COMMON /LFEEDS/ PHFBON,VFBON,USEWT
       LOGICAL PHFBON,VFBON,USEWT
C  -----------------------------------------------------------------
C
C  flowp.inc  The parameters  are from either the C or O commands.

C  The parameters governing the generation of phasespace flow contours
       DOUBLE PRECISION STVAL
       LOGICAL PLTBKT,STATIC
       COMMON/FLOWP/ STVAL(2,NKLIM)
       COMMON/IFLOWP/ LINES,KNTL(NKLIM)
       COMMON/LFLOWP/ PLTBKT,STATIC
C  -----------------------------------------------------------------
C
C  fourir.inc  (edited) Parameters not described are from the F or B commands.
C  PARAMETERS DEFINING AND RESULTS OF FOURIER TRANSFORM OF CHARGE DISTRIBUTION
C
       DOUBLE PRECISION BW,OBWGT,FAMPL,FAZE,WSAVE,F,G
       COMMON /FOURIR/ BW,OBWGT,FAMPL(NFMAX),FAZE(NFMAX),
      +    WSAVE(3*IFFT+15),F(IFFT),G(IFFT)
! BW width of bin in charge histogram for fft, FAMPL phasor amplitude,
! FAZE phase of phasor, W work space for fft, F fourier amplitudes,
! G function being analyzed
       COMMON /IFOURIR/ NBINFFT,MFFT,NNF,NBSAV,NIXNOIS,ITKNT,NF(NFMAX)
       COMMON /LFOURIR/ FFTON
```

```
      LOGICAL FFTON
C  ----------------------------------------------------------------------
C
C  gamjmp.inc  (edited) Parameters not described are from the R command.
C  Parameters defining transition gamma time dependence and
C  transition phase switch timing.
      DOUBLE PRECISION GAMPAR,ETAJMP
      COMMON /GAMJMP/ GAMPAR(5),ETAJMP
      COMMON /IGAMJMP/ KINDG
      COMMON /LGMAJMP/ GMAJMP
      LOGICAL GMAJMP
C  ----------------------------------------------------------------------
C
C  grafix.inc  (edited) Parameters not described are from the O command.
C   PARAMETERS DEFINING DESIRED GRAPHICAL OUTPUT
C
      DOUBLE PRECISION THPMIN,THPMAX,DEPMIN,DEPMAX,DTHCURV,DECURV,
     1      THBMIN,THBMAX,EBMIN,EBMAX,HBCKTN,SBCKTN,SREF,REFAREA,
     2      CVBMIN,CVBMAX,DELCON,THEXCMI,THEXCPL,DEEXCMI,DEEXCPL,SEXCL,
     3      XCRNR,YCRNR,XAXISL,YAXISL
      COMMON /GRAFIX/ THPMIN,THPMAX,DEPMIN,DEPMAX,DTHCURV,DECURV,
     1      THBMIN,THBMAX,EBMIN,EBMAX,HBCKTN,SBCKTN,SREF,REFAREA,
     2      CVBMIN,CVBMAX,DELCON,THEXCMI,THEXCPL,DEEXCMI,DEEXCPL,SEXCL,
     3      XCRNR,YCRNR,XAXISL,YAXISL
C
      COMMON /IGRAFIX/ IDEV,IMETA,MPLOT,IRF,ICONTUR,IEREF,
     1      NBINTH,NBINE,IFBMIN,IFBMAX,NPJMP,KLPLOT
C
      LOGICAL PLTSW,DRWREF,NODRAW,POSTP,LINTER,ZTABFLG,TDBFFLG
      COMMON /LGRAFIX/ POSTP,NODRAW,ZTABFLG,TDBFFLG,TITLE,PLTSW(40),
     1      DRWREF,LINTER
! DRWREF flag signals a reference contour for phase plane plot, NODRAW
! flag indicating no plotting, ZTABFLG flag to plot impedance table,
! TDBFFLG flag to plot time domain basis function, LINTER flag to pause
! between plots
C
      CHARACTER PGINFIL*254,PGOUTFIL*254
      COMMON /CGRAFIX/ PGINFIL,PGOUTFIL
! PGINFIL name of main data file, PGOUTFIL name of postscript output file
C
C  ----------------------------------------------------------------------
C
C  heading.inc
C   DESCRIPTIVE HEADING FOR GRAPHICAL OUTPUT
C
      COMMON /HEADING/ TITL
      COMMON /IHEADING/ TITLEN
      CHARACTER TITL*50
      INTEGER TITLEN
! TITLEN number of characters in plot heading
C  ----------------------------------------------------------------------
C
C     io.inc  (edited)
C     IO UNITS AND ARRAY SIZES
C
      CHARACTER*254 PGMARG, LOGFIL
! PGMARG the command line arguments, LOGFIL the ascii output file
```

```
      CHARACTER*20 TEMPFIL
! TEMPFIL prefix identifying temporary files (e.g., FORT.9 --> TEMPFIL.9)
      INTEGER KINUNIT,KOUTUNIT
! KINUNIT,KOUTUNIT unit numbers for principal input and ascii output files
      LOGICAL LCOLOR
! LCOLOR switch for color plots
C
      COMMON /IOCOML/   LCOLOR
      COMMON /IOCOMI/   KINUNIT,KOUTUNIT
      COMMON /FLAGCOM/  KVERBOSE


C
      COMMON /MEMORY/ KMAXCVB, KNPHASE, KMAXPTS
C  --------------------------------------------------------------------
C
C  mrange.inc  (edited) The parameters  are from the M command.
C  Include containing mountain range parameters for saving data;
C  the plotting parameters, however, are local to MRPLT.
      REAL MRMINB,MRMAXB,TMBEGIN,TMEND
      COMMON /MRANGE/ MRMINB,MRMAXB,TMBEGIN,TMEND
      INTEGER MRMPLOT,NRNBIN
      COMMON /IMRANGE/ MRMPLOT,NRNBIN
      LOGICAL IPU,FASPEC,FPSPEC
      COMMON /LMRANGE/ IPU,FASPEC,FPSPEC
C  --------------------------------------------------------------------
C
C  parameters.inc
C GLOBAL PARAMETER DEFINITIONS
C ---------------------------
C
#ifdef DYNMEM
C LDMEM    = .FALSE. memory allocation routine does nothing
C             except read NAMELIST and set a few integer
C             variables used in some tests
      LOGICAL LDMEM
      PARAMETER (LDMEM=.TRUE.)
#else
      LOGICAL LDMEM
      PARAMETER (LDMEM=.FALSE.)
#endif
C BCKRAT    = bucket ratio used in MATCH subroutine
      DOUBLE PRECISION BCKRAT
      PARAMETER (BCKRAT=1.26D0)
C ICURVE    = max no of points in a phase contour
      INTEGER ICURVE
      PARAMETER (ICURVE = 5001)
C KNTLIM    = maximum number of integration steps in constructing flow line or contour
      INTEGER KNTLIM
      PARAMETER (KNTLIM = 5000)
C IFBPRS    = length of circular buffer for averaging successive turns in phase feedback
      INTEGER IFBPRS
      PARAMETER (IFBPRS = 1001)
C IFFT      = maximum array size for FFT
      INTEGER  IFFT
      PARAMETER (IFFT = 8192)
C IFLSIZ    = max no of bins for Bersntein polynomial smoothing
      INTEGER  IFLSIZ
```

```
      PARAMETER(IFLSIZ=100)
C IINUNIT   =  5  INPUT  UNIT
      INTEGER IINUNIT
      PARAMETER (IINUNIT=5)
C IFINUNIT  = 32  UNIT FOR OPTIONAL INPUT FILE
      INTEGER IFINUNIT
      PARAMETER (IFINUNIT=32)
C IFLOGUNIT = 33  UNIT FOR OPTIONAL LOG FILE
      INTEGER IFLOGUNIT
      PARAMETER (IFLOGUNIT=33)
C IOUTUNIT  = 6  OUTPUT UNIT
      INTEGER IOUTUNIT
      PARAMETER (IOUTUNIT=6)
C IRLLEN    = max no of resonators in time-domain Green's func. soln.
      INTEGER IRLLEN
      PARAMETER (IRLLEN=5)
C IVERBOSE  = 0 NAMELISTS ARE NOT PRINTED
      INTEGER IVERBOSE
      PARAMETER (IVERBOSE=1)
C LTABL     = max length of a spline table
      INTEGER LTABL
      PARAMETER (LTABL=350)
C MAXBINS    = number of bins in rf voltage plot (PLTWAV)
      INTEGER MAXBINS
      PARAMETER (MAXBINS=400)
C MAXCNT    = max no of attempts to match a bucket to a bunch
      INTEGER MAXCNT
      PARAMETER (MAXCNT=12)
C MAXCVB     = max no of bins in longitudinal charge distribution
C             (used for all collective voltage calculations in time domain)
      INTEGER MAXCVB
      PARAMETER (MAXCVB=1024)
C MAXPLT    = max no of history plots
      INTEGER MAXPLT
      PARAMETER (MAXPLT=50)
C MAXPTS    = max no of points in a history plot
      INTEGER MAXPTS
      PARAMETER (MAXPTS=1000)
C MAXTRC    = max no of points on a trace in a mountain range plot
      INTEGER MAXTRC
      PARAMETER (MAXTRC=1002)
C MAXTTL    = maximum length of plot title
      INTEGER  MAXTTL
      PARAMETER (MAXTTL=50)
C NDBLSIZ   = size of a DOUBLE PRECISION, in bytes
      INTEGER NDBLSIZ
      PARAMETER (NDBLSIZ = 8)
C NFMAX     = max no of Fourier components saved for history
      INTEGER NFMAX
      PARAMETER (NFMAX = 50)
C NIT       = no of iterations for binary search (implicitly def fnct)
      INTEGER NIT
      PARAMETER (NIT=25)
C NKLIM     = max no of distinct classes of phase space point
      INTEGER NKLIM
      PARAMETER (NKLIM=50)
C NPHASE     = max no of phase points
```

```
      INTEGER NPHASE
      PARAMETER (NPHASE = 25000)
C NPMAX     = max no iterations (SYNCH) of synchronous phase for multiple RF systems
      INTEGER  MAXIT
      PARAMETER(MAXIT =50)
C NRD       = size of the random no array for Knuth's Random no generator
C                **must be > 100**
      INTEGER NRD
      PARAMETER (NRD=200)
C NREALSZ   = size of a REAL, in bytes
      INTEGER NREALSIZ
      PARAMETER (NREALSIZ = 4)
C NSPARE    = no of slots for  user-defined histories
      INTEGER NSPARE
      PARAMETER (NSPARE = 10)
C NSRC      = max no of RF systems
      INTEGER NSRC
      PARAMETER (NSRC = 10)
C NTRIALS   = no of attempts to complete a contour
      INTEGER NTRIALS
      PARAMETER (NTRIALS=6)
C NTRP      = max no of simultaneously active trapping conditions
      INTEGER NTRP
      PARAMETER (NTRP = 4)
C PVER      =  Post-processor version no
      DOUBLE PRECISION PVER
      PARAMETER (PVER=2.11)
C SSIZ      =  ? (MATCH subroutine)
      DOUBLE PRECISION SSIZ
      PARAMETER (SSIZ=5.D-1)
C TITLL     = axis label length for user-defined history plots
      DOUBLE PRECISION TITLL
      PARAMETER (TITLL=14)
C TOL       = tolerance for moving bucket factor interpolation
      DOUBLE PRECISION TOL
      PARAMETER (TOL=1.D-6)
C TOL2      = tolerance for MATCH subroutine
      DOUBLE PRECISION TOL2
      PARAMETER (TOL2=1.D-1)
C UNITL     = length of unit field in user-defined history plots
      DOUBLE PRECISION UNITL
      PARAMETER (UNITL=7)
C VERNUM    =  ESME version number
      DOUBLE PRECISION VERNUM
      PARAMETER (VERNUM=8.5)
C  -------------------------------------------------------------------
C
C ===================================================================
C
C--PARAMETERS USED IN PLOTTING ROUTINES

C     COLOR     = .FALSE.
      LOGICAL COLOR
      PARAMETER (COLOR=.TRUE.)
C
C     Viewport margins in normalized coordinates
C
```

```
        REAL VPLFT, VPRGT, VPBTM, VPTOP
        PARAMETER (VPLFT=0.10, VPRGT=0.90, VPBTM=0.10, VPTOP=0.75)
C       Plot title character size, sub-title character size,
C       Notes character size
        REAL CTSIZ,CSSIZ,CNSIZ
        PARAMETER (CTSIZ=1.5,CSSIZ=1.0,CNSIZ=0.70)
C
C ======================================================================
C
C Finally, some useful and not so useful definitions ...
C
        DOUBLE PRECISION  BIG,SMALL
        PARAMETER (BIG=9.D37, SMALL=0.7D-38)
        DOUBLE PRECISION  VYBIG,VYSMALL
        PARAMETER (VYBIG=1.7D38, VYSMALL=0.3D-38)
        DOUBLE PRECISION ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,TEN
        PARAMETER (ZERO=0.D0, ONE=1.D0, TWO=2.D0, THREE=3.0D0, FOUR=4.0D0,
     1       FIVE=5.D0, SIX=6.D0, TEN=1.D1)
        DOUBLE PRECISION ONEPT6,PT52
        PARAMETER (ONEPT6=1.6D0, PT52=0.52D0)
        DOUBLE PRECISION  HALF
        PARAMETER (HALF=5.D-1)
        DOUBLE PRECISION EIGHTEEN
        PARAMETER (EIGHTEEN=1.8D+1)
        DOUBLE PRECISION TENTO6, HALFMIL
        PARAMETER (TENTO6=1.D+6,HALFMIL=5.D+5)
C
C ======================================================================
C
C  pgplot.inc  (edited)
C----------------------------------------------------------------------
C PGPLOT: common block definition.
C----------------------------------------------------------------------
C Maximum number of concurrent devices (should match GRIMAX).
C----------------------------------------------------------------------
        INTEGER PGMAXD
        PARAMETER (PGMAXD=8)
C----------------------------------------------------------------------
C Identifier of currently selected device.
C----------------------------------------------------------------------
        INTEGER PGID
C----------------------------------------------------------------------
C Device status (indexed by device identifier).
C----------------------------------------------------------------------
C PGDEVS  =0 if device is not open; 1 if device is open.
C PGADVS  Set to 0 by PGBEGIN, set to 1 by PGPAGE; used to suppress
C         the prompt for the first page.
C PROMPT  If .TRUE., ask user before clearing page; set by PGASK
C         and (indirectly) by PGBEGIN, used in PGENV.
C PGBLEV  Buffering level: incremented by PGBBUF, decremented by
C         PGEBUF.
C PGPFIX  TRUE if PGPAP has been called, FALSE otherwise.
C
        INTEGER PGDEVS(PGMAXD), PGADVS(PGMAXD), PGBLEV(PGMAXD)
        LOGICAL PGPRMP(PGMAXD), PGPFIX(PGMAXD)
C----------------------------------------------------------------------
C Panel parameters (indexed by device identification).
```

```
C-----------------------------------------------------------------------
C NX      Number of panels in x direction
C NY      Number of panels in y direction
C NXC     Ordinal number of current X panel
C NYC     Ordinal number of current Y panel
C XSZ     X dimension of panel (device units)
C YSZ     Y dimension of panel (device units)
C PGROWS  TRUE if panels are used in row order, FALSE for column
C         order.
C
      INTEGER PGNX  (PGMAXD), PGNY  (PGMAXD)
      INTEGER PGNXC (PGMAXD), PGNYC (PGMAXD)
      REAL    PGXSZ (PGMAXD), PGYSZ (PGMAXD)
      LOGICAL PGROWS(PGMAXD)
C-----------------------------------------------------------------------
C Attributes (indexed by device identification).
C-----------------------------------------------------------------------
C PGFAS   fill-area style
C PGCHSZ  character height
C PGAHS   arrow-head fill style
C PGAHA   arrow-head angle
C PGAHV   arrow-head vent
C PGTBCI  text background color index
C PGMNCI  lower range of color indices available to PGGRAY/PGIMAG
C PGMXCI  upper range of color indices available to PGGRAY/PGIMAG
C PGITF   type of transfer function used by PGGRAY/PGIMAG
C PGHSA   hatching line angle
C PGHSS   hatching line separation
C PGHSP   hatching line phase
C
      INTEGER PGFAS (PGMAXD)
      REAL    PGCHSZ(PGMAXD)
      INTEGER PGAHS (PGMAXD)
      REAL    PGAHA (PGMAXD)
      REAL    PGAHV (PGMAXD)
      INTEGER PGTBCI(PGMAXD)
      INTEGER PGMNCI(PGMAXD)
      INTEGER PGMXCI(PGMAXD)
      INTEGER PGITF (PGMAXD)
      REAL    PGHSA (PGMAXD)
      REAL    PGHSS (PGMAXD)
      REAL    PGHSP (PGMAXD)
C-----------------------------------------------------------------------
C Viewport parameters (indexed by device identification); all are device
C coordinates:
C-----------------------------------------------------------------------
C PGXOFF  X coordinate of blc of viewport.
C PGYOFF  Y coordinate of blc of viewport.
C PGXVP   X coordinate of blc of viewport, relative to blc of subpage.
C PGYVP   Y coordinate of blc of viewport, relative to blc of subpage.
C PGXLEN  Width of viewport.
C PGYLEN  Height of viewport.
C
      REAL    PGXOFF(PGMAXD), PGYOFF(PGMAXD)
      REAL    PGXVP (PGMAXD), PGYVP (PGMAXD)
      REAL    PGXLEN(PGMAXD), PGYLEN(PGMAXD)
C-----------------------------------------------------------------------
```

```
C Scaling parameters (indexed by device identification):
C-----------------------------------------------------------------------
C PGXORG  device coordinate value corresponding to world X=0
C PGYORG  device coordinate value corresponding to world Y=0
C PGXSCL  scale in x (device units per world coordinate unit)
C PGYSCL  scale in y (device units per world coordinate unit)
C PGXPIN  device x scale in device units/inch
C PGYPIN  device y scale in device units/inch
C PGXSP   Character X spacing (device units)
C PGYSP   Character Y spacing (device units)
C
      REAL    PGXORG(PGMAXD), PGYORG(PGMAXD)
      REAL    PGXSCL(PGMAXD), PGYSCL(PGMAXD)
      REAL    PGXPIN(PGMAXD), PGYPIN(PGMAXD)
      REAL    PGXSP (PGMAXD), PGYSP (PGMAXD)
C-----------------------------------------------------------------------
C Window parameters (indexed by device identification); all are world
C coordinate values:
C-----------------------------------------------------------------------
C PGXBLC  world X at bottom left corner of window
C PGXTRC  world X at top right corner of window
C PGYBLC  world Y at bottom left corner of window
C PGYTRC  world Y at top right corner of window
C
      REAL    PGXBLC(PGMAXD), PGXTRC(PGMAXD)
      REAL    PGYBLC(PGMAXD), PGYTRC(PGMAXD)
C-----------------------------------------------------------------------
C The following parameters are used in the contouring routines to pass
C information to the action routine. They do not need to be indexed.
C-----------------------------------------------------------------------
C TRANS   Transformation matrix for contour plots; copied
C         from argument list by PGCONT and used by PGCP.
C
      INTEGER PGCINT, PGCMIN
      REAL    TRANS(6)
      CHARACTER*32 PGCLAB
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      COMMON  /PGPLT1/ PGID,PGDEVS,PGADVS,PGNX,  PGNY,  PGNXC, PGNYC ,
     1        PGXPIN,PGYPIN,PGXSP, PGYSP, PGXSZ, PGYSZ,
     2        PGXOFF,PGYOFF,PGXVP, PGYVP, PGXLEN,PGYLEN,PGXORG,PGYORG,
     3        PGXSCL,PGYSCL,PGXBLC,PGXTRC,PGYBLC,PGYTRC,TRANS,
     4        PGPRMP,PGFAS, PGCHSZ,PGBLEV,PGROWS,
     5        PGAHS, PGAHA, PGAHV, PGTBCI,PGMNCI,PGMXCI,PGCINT,PGCMIN,
     6        PGPFIX,PGITF, PGHSA, PGHSS, PGHSP
      COMMON  /PGPLT2/ PGCLAB
C-----------------------------------------------------------------------
C
C  plt.inc
C  QUANTITIES SHARED BETWEEN SUBROUTINES PLTxxx
C
#ifdef DYNMEM
#ifndef _DYNMEM_INC
#include "dynmem.inc"
#endif
      COMMON /PLTSNGL/ EPHMIN,EPHMAX,THMINP,THMAXP,DEMINP,DEMAXP,
     1                 VRFMIN,VRFMAX,BMIN,BMAX,RDEGRAD,RRADDEG
```

```
#else
      COMMON /PLTSNGL/ EPHMIN,EPHMAX,THMINP,THMAXP,DEMINP,DEMAXP,
     1                 VRFMIN,VRFMAX,BMIN,BMAX,RDEGRAD,RRADDEG,
     2                 SPHASETH,SPHASEE,BIN1,BIN2,XDHIS,PLTX,PLTY
      REAL SPHASETH(NPHASE),SPHASEE(NPHASE),PLTX(MAXPTS),PLTY(MAXPTS),
     1     BIN1(MAXCVB),BIN2(MAXCVB),XDHIS(MAXCVB)
#endif
C--Arrays used to plot line between two points
      REAL XL(2), YL(2)
C--Arrays to hold moments for histograms.
      DOUBLE PRECISION RMOM
C--Dummy arguments that will have double precision mode
      DOUBLE PRECISION ARG1,ARG2,ARG3,ARG4
C--Various notes
      CHARACTER*33  SUBTTL
      CHARACTER*11  CHRNUM(3,3)
      CHARACTER*10  CHRV,CHRPSI,CHRBNC,CHRTIM,CHRBAR,CHRRMS,FMT
      CHARACTER*32  CHRBLK,CHRSC,CHRNPP
      CHARACTER*9   CHRMM
      CHARACTER*4   CHRH
      CHARACTER*3   CHR123
      CHARACTER*6   CHVLBL1, CHVLBL2
      CHARACTER PREFIX,PFY
      CHARACTER*72 CHCLBL
      CHARACTER*14 CHELBL,CHNBIN
C--Useful switch combinations
      LOGICAL BNCHON,LINEON
C
C--Misc Declarations for PGPLOT Graphics
C
      PARAMETER (NIMAGB=250)
      REAL DENSS(NIMAGB,NIMAGB),DENS(NIMAGB,NIMAGB), TR(6)
C
      COMMON /PLTLBL/ SUBTTL,CHRNUM,
     +    CHRV,CHRPSI,CHRBNC,CHRTIM,CHRBAR,CHRRMS,FMT,
     +    CHRMM,CHRBLK,CHRSC,CHRNPP,
     +    CHRH,
     +    CHR123,
     +    CHVLBL1, CHVLBL2,
     +    PFY,
     +    CHCLBL,
     +    CHELBL,CHNBIN
C
      COMMON /PLTDBL/ RMOM(6)
C
      COMMON /PLTINT/ IPXLSIZ, LCHRMM
C  --------------------------------------------------------------------
C
C  poplate.inc  (edited) The parameters are from the P command.
C  THE PARAMETERS DEFINING THE INITIAL PHASESPACE DISTRIBUTION(S)
      DOUBLE PRECISION THMIN,THMAX,REMIN,REMAX,
     1     SBNCH,THOFF,EOFF,THTRAN,ETRAN
      COMMON /POPLATE/ THMIN,THMAX,REMIN,REMAX,
     1     SBNCH,THOFF,EOFF,THTRAN,ETRAN
      COMMON /IPOPLATE/ KIND,IPOP,NTH,NE,NPOINT
C  --------------------------------------------------------------------
C
```

```
C   random.inc
C
        DOUBLE PRECISION DLARAN
        DOUBLE PRECISION XRAND
        INTEGER ISEED
!   DLARAN is the random number generator, XRAND a quasi-random number
C
        COMMON /RNDOM/ ISEED(4)
!   ISEED the starting value for random number generator initially set in
!   esinit
C   ----------------------------------------------------------------------
C
C   rfp.inc  (edited) Parameters not described are from the A command.
C    PARAMETERS DEFINING THE RF SYSTEMS
C
        DOUBLE PRECISION HDECR,SDECR,PHISLIM,VI,VF,TVBEG,TVEND,
       1        FRI,FRF,TFBEG,TFEND,PSII,PSIF,TPBEG,TPEND,
       2        VTABL,FTABL,PTABL,C1,C2,DELTRF
        COMMON /RFP/ HDECR,SDECR,PHISLIM,
       1        VI(NSRC),VF(NSRC),TVBEG(NSRC),TVEND(NSRC),
       2        FRI(NSRC),FRF(NSRC),TFBEG(NSRC),TFEND(NSRC),
       3        PSII(NSRC),PSIF(NSRC),TPBEG(NSRC),TPEND(NSRC),
       4        VTABL(LTABL,3,NSRC),FTABL(LTABL,3,NSRC),PTABL(LTABL,3,NSRC),
       5        C1(NSRC),C2(NSRC),DELTRF(NSRC)
!   VTABL spline table of rf voltages vs. time, FTABL rf frequencies vs.
!   time table, PTABL rf phases vs. time, C1,C2 constants in iso-adiabatic
!   voltage curves
        INTEGER H,HW,HGCD,HMAX
        COMMON /IRFP/ NRF,H(NSRC),HW(NSRC),HGCD,HMAX,ISYNC,
       1        KURVE(NSRC),NTV(NSRC),NTABV(NSRC),
       2        KURVF(NSRC),NTF(NSRC),NTABF(NSRC),
       3        KURVP(NSRC),NTP(NSRC),NTABP(NSRC)
!   HGCD the greatest common divisor of the rf system harmonic numbers,
!   HMAX the highest harmonic number, NTV,NTF,NTP number of entries in voltage
!   table, frequency table, and phase tables respectively
!   NTABV,NTABF,NTABP index of the last used entry in the voltage, frequency,
!   and phase tables respectively
        LOGICAL VKON,FRKON,PHKON,PHSLIP,VMATCH
        LOGICAL HOLDBH,HOLDBA,CNTINU
        COMMON /LRFP/ VKON,FRKON,PHKON,PHSLIP,HOLDBH,HOLDBA,CNTINU(NSRC),
       1        VMATCH(NSRC)
C   ----------------------------------------------------------------------
C
C   ringp.inc  (edited) Parameters not described are from the R command.
C    THE LATTICE PARAMETERS INCLUDING TIME DEPENDENCES
        DOUBLE PRECISION REQ,GAMTSQ,ALPHA0,ALPHA1,ALPHA2,ALPHA3,TAUINF,
       1        W0I,W0F,EK0I,EK0F,TI,TF,P0I,P0F,PI,PF,FRAC,DES,PIPRAD,
       2        W0IDOT,W0FDOT,EKIDOT,EKFDOT,P0IDOT,P0FDOT,PIDOT,PFDOT,
       3        THLO,THHI,THRNG,CRA,CRB,CRC,TR3,CHGNO,EM0CSQ,EMCSQ
        COMMON /RINGP/ REQ,GAMTSQ,ALPHA0,ALPHA1,ALPHA2,ALPHA3,TAUINF,
       1        W0I,W0F,EK0I,EK0F,TI,TF,P0I,P0F,PI,PF,FRAC,DES,PIPRAD,
       2        W0IDOT,W0FDOT,EKIDOT,EKFDOT,P0IDOT,P0FDOT,PIDOT,PFDOT,
       3        THLO,THHI,THRNG,CRA,CRB,CRC,TR3,CHGNO,EM0CSQ,EMCSQ
!   THLO bottom of calculation range -pi/FRAC, THHI top of calculation range
!   pi/FRAC, THRNG calculation range 2 pi/FRAC, EMCSQ rest energy of beam particles
        COMMON /IRINGP/ KURVEB,NCAV
        LOGICAL JNRAMP,EBDRY
```

```
      COMMON /LRINGP/ JNRAMP,EBDRY
C  ----------------------------------------------------------------------
C
C  spares.inc
C  Contains "spare" variables which are written to history
C  along with everything else.  These can be whatever the user
C  wishes; these commons are available to CYCPROG and SHAZAM.
C  Also contains labels and units for spares used in history plots.
C
      DOUBLE PRECISION SPARE
      COMMON /SPARES/ SPARE(NSPARE)
      COMMON /CSPARES/ SPLABL(NSPARE),SPUNIT(NSPARE)
      CHARACTER*14 SPLABL
      CHARACTER*7 SPUNIT
C  ----------------------------------------------------------------------
C
C  spchg.inc  (edited) Parameters not described are from the B command.
C  PARAMETERS DEFINING THE SPACE CHARGE & WALL IMPEDANCE ENERGY/TURN
c
#ifdef dynmem
c
      DOUBLE PRECISION A,B,ENQ,TCHGON,BWSC,TDELTA,ZTABL,EVFD,
     1      RESTBL,RV0,RV0DOT
c
#ifndef _DYNMEM_INC
#include "dynmem.inc"
#endif
      COMMON /SPCHG/ A,B,ENQ,TCHGON,BWSC,TDELTA,ZTABL(LTABL,5),
     1      EVFD(IFFT),RESTBL(IRLLEN,4),RV0(IRLLEN),RV0DOT(IRLLEN)
c
#else
c
      DOUBLE PRECISION A,B,ENQ,TCHGON,BWSC,TDELTA,ZTABL,EVFD,RESTBL,
     1      ZRES,EVRES,RV0,RV0DOT,DQ,THTA,DQ2,TDTBL,FIT,EVTD
      COMMON /SPCHG/ A,B,ENQ,TCHGON,BWSC,TDELTA,ZTABL(LTABL,5),
     1      EVFD(IFFT),RV0(IRLLEN),RV0DOT(IRLLEN),RESTBL(IRLLEN,4),
     2      ZRES(MAXCVB),EVRES(MAXCVB),DQ(MAXCVB),THTA(MAXCVB),
     3      DQ2(MAXCVB),TDTBL(MAXCVB),FIT(MAXCVB),EVTD(MAXCVB),
     4      DQDT(MAXCVB)
#endif
! BWSC width of bins in charge histogram, TDELTA time step from time domain
! basis table file, ZTABL impedance vs frequency spline table, EVFD collective
! voltage calculated in frequency domain, RESTBL table of simple resonances,
! EVRES collective voltage from special time domain resonance calculation,
! TDTBL time domain basis function, EVTD collective voltage from time domain
! calculation.
!! Variables not identified here or in command parameter table are intermediate
!! result tables  --- no touchee!
C
      COMMON /ISPCHG/ NZ,NR,NT,MSC,NBINSC,NBRES,NBINTD
      LOGICAL  SCON,FDON,FDSCON,QREZON,TDON
! NZ no of entries in impedance table, NR number of entries in resonance table,
! NT no of entries in time domain basis table
      COMMON /LSPCHG/ SCON,FDON,FDSCON,QREZON,TDON
C  ----------------------------------------------------------------------
C
C  times.inc
```

```
C  A BLOCK TO CONTAIN CPU TIME SINCE START
      COMMON/TIMES/ CPUBEG,CPUNOW
      COMMON/ITIMES/ ITIME,IRCOD1,IRCOD2
      DOUBLE PRECISION CPUBEG, CPUNOW
C  ----------------------------------------------------------------------
C
C  version.inc
C  VERNUM is the version of ESME
C  PVER is the post-processor version number.
C  These are parameters given in the parameters.inc include file.
C  Version 2001 ESME2000 (ESME for the millenium, Y2K compliant)
C
C  alpha_i treated consistently as delta-R/R expansion coefficients (v. 7.2)
C  Contours for accelerating/decelerating below/above transition OK (v. 7.31)
C  BUCKIT tries starting points relative to both fixed points (v. 7.32)
C  Provision for generating flow contours including collective pot. (v. 8.00)
C  Improved treatment of collective potential (v. 8.03)
C  As documented in TM-1835 (v. 8.04)
C  Update for HIGZ graphics et al. documented in TM-1844 - v. 8.05
C  Misc cleanup for diverse platforms; additions to R, F, B, & M cmds documented
C  in TM-1856, a re-issue of the User's Guide 26 Aug 93 - v. 8.10
C  25 July 94 - Format for complex impedance changed to f, z_r, z_i (v. 8.12)
C             - Interpolation routine for LINBKT (INTPGAM) strengthened
C  18 Aug 94  - Voltage feedback (LOWLVL, CYCPROG) fixed for ramping (v. 8.13)
C  13 Nov 96  - Removing DI3000 and single precision options
C             - adding some distributions with low numerical noise
C  23 Oct 97  - beginning of major overhaul including time domain space charge
C               arbitrary particle species, and much else - called ESMY2K
C  31 Dec 98  - ESME-2000 New features, new documentation, data incompatibilities
C  05 Aug 00  - v.2001 Scaling for speed and more dynamic allocation plus etc.
C Note on authorship - The code has been written primarily by James MacLachlan of
C Fermilab starting in the early 1980's. Important contributions have also been made by
C   Peter Lucas - 1986        for Version 6 and DI-3000 graphics
C   Stephen Stahl - 1987-92   for Version 7 and HIGZ graphics
C   Francois Ostiguy - 1996-98 for Version 8.2 and pgplot graphics &
C                                    run-time memory allocation
```

4.2 Makefile and Compilation Switches

For Unix systems there exists a Makefile which allows the code to be compiled in optional forms. It uses source files that have extensions of .F for program and .inc for include files to be introduced by the C preprocessor. The existing source has preprocessor directives for compiling an optional map where long-term numerical stability is an issue and for Sun and AIX system dependencies. The Makefile is easily modified for debugging or optimized compilation and linking. A representitive version follows:

```
SHELL = /bin/sh
VPATH = ../
# ---------------------------------------------------
# Makefile for ESME  SUN4c architecture PGPLOT
# ---------------------------------------------------
#
# June 1997
#
# In case of difficulty please visit
#
# http://www-ap.fnal.gov/ESME
#
```

```
# for latest information, patches etc ...
#
# or contact:
#
# James MacLaclan  AD/MI Fermilab maclachlan@fnal.gov
# Francois Ostiguy AD/AP Fermilab ostiguy@fnal.gov
#
# -----------------------------------------------------
#
# ****************************
# Comment/uncomment as needed.
# ****************************
#
# The C preprocessor command
#
CCPP     = /usr/ccs/lib/cpp -P
#
# The X11 libraries
#
X11LIBS   =-L/usr/local/ap/X11R6/lib -lX11 -lsocket -lnsl -R/usr/local/ap/X11R6/lib
#
# The PGPLOT library
#
PGPLOTLIB = /usr/local/ap/lib/libpgplot.a
#
# SUN       define for SUN
# DYNMEM    define to enable dynamic memory allocation
# DBG       define for debugging. Turns off buffering fort plots.
# CVCALC    enable calculation of collective potential (TD or FD)
# BNDRY     enable boundary tests in inner loop
# MODDEQ    use modified map instead of phase angle change at transition
# -------------------------------------------------------
# SUNOS 5.X  (SOLARIS) with SUN f77 Compiler
# -------------------------------------------------------
F77 = f77
#DEFS=  -DSUN
#DEFS=  -DSUN  -DDYNMEM
#DEFS=  -DSUN  -DDBG -DDYNMEM
#DEFS=  -DSUN -DDYNMEM -DCVCALC -DBNDRY
DEFS=  -DSUN -DDDBUG -DCVCALC -DBNDRY
FFLAGS  = -g -C
#FFLAGS = -fast  -xarch=generic                     # optimized for generic SPARC
LDFLAGS= -g -C
#LDFLAGS=-v -fast -xarch=generic -xlibmopt          # optimized math library
#
# =========================================================
# YOU SHOULD NOT HAVE TO CHANGE ANYTHING BELOW THIS LINE !
# =========================================================

OBJDIR = ./
FCOMPILE = $(F77) -c $(DEFS) $(FFLAGS)
LINK = $(F77) $(FFLAGS) $(LDFLAGS)

bin_PROGRAMS = esme

# Source files for new PGPLOT graphics
#
```

```
GRAPHSRC  = drawinit.F esinit.F   esmain.F   esquit.F    \
            exclbnd.F  grafset.F  history.F  mrplt.F     \
            phplt.F    pltblv.F   pltcntur.F pltdens.F   pltenrgy.F  \
            pltfdvc.F  pltfou.F   pltphase.F pltresn.F   \
            pltspchg.F plttdbf.F  plttdvc.F  pltthta.F   \
            pltwav.F   pltztab.F  select.F   \


esme_srcs = beamsc.F   bernst.F   bfunct.F   blchg.F        blockdata.F \
            bnch.F     buckit.F   chgdist.F  contour.F   \
            cycprog.F  delesc.F   display.F  dlaran.F    \
            dtabout.F  engfmt.F   fampout.F  famphst.F   \
            fdvcoll.F  fft.F      ffti.F     fftset.F    \
            fillbe.F   fillbfg.F  fillbg.F   \
            fillbp.F   fillbpr.F  fillbr.F   fillbu.F    \
            fillfix.F  floline.F  flowprg.F  fourr.F     \
            freqnc.F   gcd.F      get.F      getdat.F    \
            glabl.F    hiqres.F   intpgam.F  inverf.F    \
            karve.F    lentrue.F  linbkt.F   loop.F      \
            lost.F     lowlvl.F   match.F    mbfctr.F    \
            memalloc.F memfree.F  moments.F  mrinit.F    \
            mrsave.F   outlb.F    outside.F  phasflo.F   \
            point.F    popul8.F   prefix.F   prntout.F   \
            refcont.F  rfprog.F   ringpar.F  \
            rmoments.F rootf.F    rtangl.F   save.F      \
            septrix.F  shazam.F   spline.F   synch.F     \
            tablout.F  tdvcoll.F  topbtm.F   trap.F      \
            volts.F    wght.F     ztabout.F  ztabspl.F   \
            ${GRAPHSRC}

esme_objs   = ${esme_srcs:.F=.o}
esme_src2   = ${esme_srcs:.F=.f}


.SUFFIXES:


.SUFFIXES: .c .f .F .o


.F.f:
$(CCPP) $(DEFS) $< > ${*F}.f
.f.o:
$(FCOMPILE) $*.f

esm2k:  $(esme_objs) $(esme_src2)
 $(LINK) -o esm2k $(esme_objs) \
 $(PGPLOTLIB) $(X11LIBS)


clean:
/bin/rm *.o *.f esm2k


.NOEXPORT:
```

This example will probably fit most needs but is not a supported item. Individual circumstances are likely to require some modifications. Notice the optional compilation for certain features of the code that involve IF tests in the innermost tracking loop. When running on a pipelined machine, there can be some gain in efficiency in removing these tests if the data is such that they are always negative, *i. e.*, if the corresponding feature is not being used. The use of the alternative single particle map is discussed in Sec. 1.2.

Acknowledgement (J. MacLachlan)

Appendix A

New and Obsolete Input Parameters

It is convenient to go back to old data sets which are close to what is needed for a new case, but the input variables have changed somewhat over the years. Nonetheless, the changes needed to modernize even fifteen-year-old data are not difficult so long as one knows what has changed. The first table in this appendix lists all the variables which have become obsolete since v. 7.1 along with a replacement if there is one. The second table gives new variables which have no close antecedent. A look at these tables may suffice to give all the information needed for resurrecting ancient data.

Table A.1: Obsolete variables and their replacements since v. 7.1

Command	Obsolete name	v. 2001 name	Obsolete name	v. 2001 name
A	EXCHRF		VMATCHF	
	VMATCHI	VMATCH		
B	NR	(on file)	TSCON	TCHGON
	NZ	(on file)	MAXFFTB	NBINFFT
	SCON	various		
C	DELMIN		DELMAX	
M	MRTHBMIN	MRBMIN	MRTHBMAX	MRBMAX
N	MRTHPMIN	MRPMIN	MRTHPMAX	MRPMAX
	SMOOTH (*logical*)	SMOOTH (*integer*)		
O	TITLE (*logical*)	TITL (*string*)	RBMIN	CVBMIN
	IOPT	IRF	RBMAX	CVBMAX
	MTCHSI	ICONTUR (=2)	MTCH95	ICONTUR (=4)
	SCBMIN	CVBMIN	SCBMAX	CVBMAX
P	ISEED		SEED	
R	EK0I	W0I	EKIDOT	W0IDOT
	EK0F	W0F	EKFDOT	W0FDOT
	GAMMAT	(GAMTSQ)		
T	MOMNTS		NCAV	(in **R** command)
Y	KMAXRB	KMAXCVB		

Table A.2: New variables since v. 7.1

Command				
A	FILCRV			
B	FILRES	FILIMP	FILTDB	NBINSC
	SCON	TDON	FDON	FDSCON
C	STVAL	STATIC	PLTBKT	
H	NWRT	IMETA	XCRNR	YCRNR
	XAXISL	YAXISL		
L	BLFBON	BLFFON	ICAV	ENQ
	RSHUNT	QCAV	DTUNE	GAIN
	DELAY	FTOP	WDTHFCTR	CHGFCTR
	NHALF	FILBWD		
M	IPU	FASPEC	FPSPEC	
N	OBJWGT	ITNO	IMETA	XCRNR
	YCRNR	XAXISL	YAXISL	
O	THEXCMI	THEXCPL	DEEXCMI	DEEXCPL
	SEXCL	ICONTUR	IMETA	XCRNR
	YCRNR	XAXISL	YAXISL	
P	RENORM			
R	CRA	CRB	CRC	TR3
	P0I	P0F	P0IDOT	P0FDOT
	NCAV	JNRAMP	GAMTSQ	
T	RSCALE0	HISTSIZ		
Y	KNPHASE	KMAXPTS		

Appendix B

Post-Processing

For those users who wish to process ESME data independently of the program, a post-processor option is provided.[1] When this option is in effect, any plotting routine calls are substituted for by writes of ESME's common blocks (containing essentially all of the information about the current state of the simulation) to FORTRAN unit 18 using subroutine SAVE. Later, this file may be read using subroutine GET. The code for a graphics post-processor is appended here as an example. The plotting routines employed here are the same ones imbedded in ESME. This graphics driver is modelled after the main program. As in ESME, one-letter commands initiate various routine calls and namelist reads. Since the plotting routines are those of ESME, for which the graphical output options set in the **O** Command were specifically intended, the user can construct plots using those options set during the running of the program as retrieved from COMMON. Those employing other graphics routines may wish to implement another sort of interface entirely (e.g. menu-driven), with an entirely different set of output options.

```
      PROGRAM ESMEPLOT
C  This program "post-processes" ESME output data.  It may read and plot
C  history data or process the output file generated by ESME during a run.
C
C  Updated to ESME v. 8.0 26-Feb-93 -- J. MacLachlan
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)

      INCLUDE 'grafix.inc'
      INCLUDE 'curves.inc'
      INCLUDE 'version.inc'
      DATA NODRAW,POSTP /.TRUE.,.FALSE./
      DATA IDEV /1/
      CHARACTER*1 CMND
      CHARACTER*74 WORDS
      LOGICAL EXFG, DONE
      CHARACTER*128 GRAFIL,HFIL,MRFIL,OHFIL,OGRAFIL,OMRFIL
      DATA GRAFIL,HFIL,MRFIL /3*' '/
C
      PRINT 19,PVER
19    FORMAT(//1X,'ESME RF PLOTTING PROGRAM : VERSION NUMBER IS ',F5.2)
C
C  Return point after execution of a command.
   10 CONTINUE
C  READ SINGLE LETTER COMMANDS.
      PRINT *,'ENTER COMMAND: '
      READ(5,2000) CMND,WORDS
 2000 FORMAT(A1,4X,A74)
       PRINT 2020, CMND,WORDS
 2020 FORMAT(' CMND IS ',A1,'   ',A74)
```

---

[1] POSTP=T, **O** Command, Section 2.2.4.

```
      GO TO (100,200,300,400,500,600,700,800,900)
     +       INDEX('ODGHZXUNQ',CMND)
      PRINT*, 'THE IMPLEMENTED COMMANDS ARE'
      PRINT*, '     O: SET OUTPUT OPTIONS'
      PRINT*, '     D: DISPLAY PHASE SPACE WITH CURRENT PLOT PARAMETERS'
      PRINT*, '     G: GET A RECORD FROM TAPE 7'
      PRINT*, '     H: RETRIEVE AND PLOT HISTORY DATA FROM TAPE 9'
      PRINT*, '     Z: REWIND TAPE 7'
      PRINT*, '     X: PLOT REMAINDER OF TAPE 7'
      PRINT*, '     U: CHOOSE GRAPHICS AND/OR HISTORY FILES'
      PRINT*, '     N: COMPOSE AND DISPLAY MOUNTAIN RANGE PLOT'
      PRINT*, '     Q: STOP'
C  Stop if the command wasn't recognized.
      STOP
C
C  O: OPTIONS FOR OUTPUT GRAPHICS
C
  100 CALL GRAFSET
      GO TO 10
C
C  D:  DISPLAY GRAPHICALLY PHASE POINTS AND (OPTIONALLY) BUCKET
C
  200 CONTINUE
      IF(ICONTUR.GT.0) CALL REFCONT
      DO 210 I=1,NC
        CURVE(I,1)=CURVE(I,1)+DTHCURV
        CURVE(I,2)=CURVE(I,2)+DECURV
  210 CONTINUE
C  Plot distribution.
      CALL PHPLT
      GO TO 10
C
C  G: GET COORDINATES AND MACHINE PARAMETERS
C
  300 PRINT *, 'READING COORDINATES AND MACHINE PARAMETERS'
C  Save logical telling us whether we have initialized
C  graphics package or not and reset it after restoring COMMONs.
      SAVND=NODRAW
      CALL GET(.TRUE.,ITURN,DONE)
      NODRAW=SAVND
      POSTP=.FALSE.
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      GO TO 10
C
C  H: HISTORY OF IMPORTANT PARAMETERS COLLECTED ON TAPE9 EACH STEP
C
  400 CALL HISTORY(.TRUE.)
      GO TO 10
C
C  Z: REWIND GRAPHICAL OUTPUT TAPE TO BEGINNING (NOW THE
C     ONLY WAY TO GET AT A RECORD BEFORE THE CURRENT ONE)
C
  500 REWIND(7)
C  Save logical telling us whether we have initialized
C  graphics package or not and reset it after restoring COMMONs.
      SAVND=NODRAW
      CALL GET(.TRUE.,ITURN,DONE)
```

```fortran
      NODRAW=SAVND
      POSTP=.FALSE.
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      GO TO 10
C
C  X: JUST PLOT EVERYTHING FROM HERE ON OUT
C
  600 CONTINUE
C  Save logical telling us whether we have initialized
C  graphics package or not and reset it after restoring COMMONs.
      SAVND=NODRAW
      CALL GET(.TRUE.,ITURN,DONE)
      NODRAW=SAVND
      IF(DONE) THEN
        PRINT *,'ALL PLOTS COMPLETED'
        GO TO 10
      ENDIF
      PRINT *, 'DATA FOR TURN ',ITURN,' READ'
      IF(ICONTUR.GT.0) CALL REFCONT
      IF(DTHCURV.EQ.0 .AND. DECURV.EQ.0) GO TO 620
      DO 610 I=1,NC
        CURVE(I,1)=CURVE(I,1)+DTHCURV
        CURVE(I,2)=CURVE(I,2)+DECURV
  610 CONTINUE
  620 CALL PHPLT
      GO TO 600
C
C  Use specified graphics and history files.
C
  700 CONTINUE
      OGRAFIL=GRAFIL
      OHFIL=HFIL
      OMRFIL=MRFIL
      NAMELIST /USEFIL/ GRAFIL, HFIL, MRFIL
      READ(5,USEFIL)
      IF(GRAFIL .NE. OGRAFIL) THEN
C  First close old file if this is a new one
        IF(OGRAFIL.NE.' ') CLOSE(UNIT=7)
C  See if the requested file exists.
        INQUIRE(FILE=GRAFIL,EXIST=EXFG)
C  Exit if it doesn't
        IF(.NOT.EXFG) THEN
          PRINT *,'FILE NAMED ',GRAFIL,' CANNOT BE FOUND '
          GOTO 710
        ENDIF
C  Print reassurance if it does
        PRINT *,'GRAPHICAL DATA FROM FILE',GRAFIL
C  Open the requested file.  Note that we haven't read
C  anything yet.
        OPEN(UNIT=7,FILE=GRAFIL,ACCESS='SEQUENTIAL',
     1        STATUS='OLD')
      ENDIF
  710 CONTINUE
      IF(HFIL .NE. OHFIL) THEN
C  First close any old history file.
        IF(OHFIL.NE.' ') CLOSE(UNIT=9)
C  See if the requested file exists.
```

```fortran
            INQUIRE(FILE=HFIL,EXIST=EXFG)
C  Exit if it doesn't
            IF(.NOT.EXFG) THEN
              PRINT *,'FILE NAMED ',HFIL,' CANNOT BE FOUND '
              GO TO 720
            ENDIF
C  Print reassurance if it does
          PRINT *,'HISTORY DATA FROM FILE',HFIL
C  Open the requested file.
            OPEN(UNIT=9,FILE=HFIL,ACCESS='SEQUENTIAL',
     1       STATUS='OLD')
          ENDIF
  720 CONTINUE
          IF(MRFIL .NE. OMRFIL) THEN
C  First close old file if this is a new one
          IF(OMRFIL.NE.' ') CLOSE(UNIT=20)
C  See if the requested file exists.
            INQUIRE(FILE=MRFIL,EXIST=EXFG)
C  Exit if it doesn't
            IF(.NOT.EXFG) THEN
              PRINT *,'FILE NAMED ',MRFIL,' CANNOT BE FOUND '
              GOTO 730
            ENDIF
C  Print reassurance if it does
            PRINT *,'GRAPHICAL DATA FROM FILE',MRFIL
C  Open the requested file.  Note that we haven't read
C  anything yet.
            OPEN(UNIT=20,FILE=MRFIL,ACCESS='SEQUENTIAL',
     1         STATUS='OLD')
          ENDIF
  730 CONTINUE
        GO TO 10
C
C  M: DISPLAY MOUNTAIN RANGE(s)
C
  800 CONTINUE
        CALL MRPLT
        GO TO 10
C
C  Q: QUIT PROGRAM ENTIRELY; NOTHING FURTHER TO DO
C
  900 CONTINUE
        PRINT *, 'QUIT COMMAND'
C
        IF(.NOT.NODRAW) CALL .......   ! Close graphics files
        STOP
        END
```

Appendix C

Command Table Summary

	Default		
Variable	Value	Unit	Description
NRF	1	-	Number of active RF sources
H(1:10)	1	-	Harmonic numbers of sources (integers)
HW(1:10)	1	-	Voltage sources will be expressed over the ϑ range $-180°/\text{HW} \leq \theta \leq 180°/\text{HW}$ to produce isolated buckets or barriers
ISYNC	0	-	Indicates synchronism condition to be imposed on RF: 0 – None, voltages and phases remain as programmed 1 – Phase of RF waveform shifted to synchronous, stable point 2 – Magnitude of RF waveform scaled to give correct synchronous energy gain 3 – Source 2 Landau cavity to source 1, synchronism assured only for sources 1 and 2
VI(1:10)	0.0	MV	Voltage of source I at time TVBEG(I)
VF(1:10)	0.0	MV	Voltage of source I at time TVEND(I)
TVBEG(1:10)	0.0	s	Time corresponding to beginning of RF voltage change
TVEND(1:10)	0.0	s	Time corresponding to end of RF voltage change
KURVE(1:10)	0	-	Specifies type of RF voltage variation between times TVBEG and TVEND: 0 – None, voltage maintained at VI(I) 1 – Linear 2 – Iso-adiabatic 3 – Sigmoid 4 – Fit and cubic spline interpolation of voltage table[a]
VKON	T	-	Indicates whether programmed voltage curves are to be active
PSII(1:10)	0	deg	Phase of source I at time TPBEG(I)
PSIF(1:10)	0	deg	Phase of source I at time TPEND(I)
TPBEG(1:10)	0.0	s	Time corresponding to beginning of RF phase change
TPEND(1:10)	0.0	s	Time corresponding to end of RF phase change
KURVP(1:10)	0	-	Specifies type of RF phase variation between times TPBEG and TPEND: 0 - None, phase maintained at PSII(I) 1 - Linear 2 - Quadratic 4 - Fit and cubic spline interpolation of phase table[a]
PHKON	F	-	Indicates whether or not phase curves are to be active
FRI(1:10)	0.0	MHz	Frequency of source I at time TFBEG(I)
FRF(1:10)	0.0	MHz	Frequency of source I at time TFEND(I)
TFBEG(1:10)	0.0	s	Time corresponding to beginning of frequency change
TFEND(1:10)	0.0	s	Time corresponding to end of frequency change
KURVF(1:10)	0	-	Specifies type of frequency variation between times TFBEG and TFEND: 0 - None, frequency maintained at FRI(I) 1 - Linear 2 - Quadratic 4 - Fit and cubic spline interpolation of frequency table[a]
FRKON	F	-	Indicates whether frequency curves are to be active
FILCRV	'DUMMY'	-	Full path for rf curves file
CNTINU(1:10)	F	-	Sets the starting voltage, phase and/or frequency for the corresponding sources to the current values — for smoothly piecing curve segments together
VMATCH(1:10)	F	-	For source I, $\text{VMATCH}(I) = T$ results in the VI(I) being set so that source I is matched to the current distribution emittance [b]
HOLDBH	F	-	Switch to calculate the voltage of *principal source*[c] to hold the current bucket height. (see also HDECR)
HDECR	1.0	-	Factor by which bucket height for *principal source*[c] is to be adjusted on successive turns if HOLDBH = T
HOLDBA	F	-	Switch to calculate the voltage of *principal source*[c] to hold the current bucket area. (see also SDECR)
SDECR	1.0	-	Factor by which bucket area for *principal source*[c] is to be adjusted on successive turns if $\text{HOLDBA} = T$
PHISLIM	.95	-	Voltage may not be reduced such that $\sin\phi_s > \text{PHISLIM}$ using options HOLDBH and HOLDBA
PHSLIP	F	-	Switch indicating that the phase of at least one source is to be varied to correspond to a momentum offset from the synchronous value (see DELTRF)
DELTRF(1:10)	0.0	-	Momentum offset $(\Delta p/p)$ at which source I is to be operated

[a]Fit to values read from file. See Section 3.2.2.

[b]Which means, in this instance, that a P command, or its equivalent, should precede the A command.

[c]The constant bucket area or bucket height is calculated from the rf source which gives the greatest bucket height.

B Command, Namelist /SCHG/

Variable	Default Value	Unit	Description
A	0.005	m	Effective beam radius
B	0.05	m	Effective beampipe radius
ENQ	$2. \cdot 10^{10}$	-	Number of protons to be represented by the distribution
NBINSC	100	-	Number of bins for histogram of charge distribution
MSC	1	-	Collective effects are to be calculated MSC times between rf cavities[a]
TCHGON	0.	s	End of period starting at TIME = 0. in which beam charge is ramped linearly from 0. to ENQ
SCON	F	-	Activate time domain perf. cond. wall calculation
TDON	F	-	Activate time domain wake field calculation with supplied response function[b]
FDON	F	-	Activate frequency domain wall impedance calculation
FDSCON	F	-	Activate frequency domain calculation of perf. cond. wall voltage
QREZON	F	-	Activate time domain calculation for high-Q resonance
NBINFFT	256	-	Number of bins to be used in Fourier transform
MFFT	1	-	Interval (in turns) between Fourier transforms
NNF [c]	0	-	Number of fourier harmonics to be stored in history
NF(1:NNF)[c]	0	-	Harmonic numbers of fourier spectrum components to be stored
NBRES	1000	-	Number of time slices for time domain solution of high-Q resonator
NIXNOIS	0	-	Three-way switch to control smoothing of charge distribution -1 => 1-2-1 averaging of adjacent bins 0 => no smoothing measures 1 => Bernstein polynomial smoothing
ITKNT	1	-	Number of iterations for either 1-2-1 or Bernstein smoothing
OBWGT	0.05	-	Weight of fitting vs. smoothing in object function for Bernstein smoothing
FILIMP	'DUMMY'	-	Full path for file containing the impedance table
FILRES	'DUMMY'	-	Full path for file containing the resonance list
FILTDB	'DUMMY'	-	Full path for file containing time domain response basis

[a]The number of such calculations per turn will be MSC*NCAV; NCAV is a parameter of the **R** command.

[b]see Appendix 3.2.6

[c]For instruction in the use of NF(1:NNF), see the description following the **F** command.

C Command, Namelist /FLOW/

Variable	Default Value	Unit	Description
LINES	2	-	The number of flow lines to be drawn
STVAL	0.,0.	[a]	LINES ϑ, E starting values for contours
STATIC	.TRUE.	-	Contours drawn with parameters fixed at initial values for STATIC = .TRUE.; parameters vary according to their programs for STATIC = .FALSE.
PLTBKT	.TRUE.	-	Option switch to include bucket contour on flow line plot
TSTOP[b]	0.	s	End of interval over which maps are generated
TTRACK[c]	0.	s	Time over which periodic flow maps will be generated
ACCEL0	1.0	-	Number of beam turns per step between mappings
PARTION	F	-	Switch indicates if points on separate flow lines belong to different partitions of the phase points
ITRAP(1:4)	1	-	Flags a condition for which the **C** command should be interrupted; see **T** command for values
ETATRP	.001	-	Trapping parameter; see **T** command
PHISTRP	.95	-	Trapping parameter; see **T** command
MGRACE	0	-	Trapping parameter; see **T** command

[a]Starting values for contours have units degrees,MeV.

[b]TSTOP set to 0. when calculation complete or interrupted by an ITRAP option.

[c]TSTOP takes precedence; if TSTOP=0., TTRACK determines duration.

F Command, Namelist /FFT/

Variable	Default Value	Unit	Description
FFTON	F	-	Activate Fourier transform calculation
FFTOUT	F	-	If TRUE, Fourier transform is printed
NBINFFT	256	-	Number of bins to be used in FFT
NNF	0	-	Number of fourier harmonics to be stored in history
NF(1:NNF)	0	-	Harmonic numbers of fourier spectrum components to be stored
MFFT	1	-	Frequency of Fourier transform calculation
NIXNOIS	0	-	Three-way switch to control smoothing of azimuthal histogram -1 => 1-2-1 averaging of adjacent bins 0 => no smoothing measures 1 => Bernstein polynomial smoothing
ITKNT	1	-	Number of iterations for either 1-2-1 or Bernstein smoothing
OBWGT	0.05	-	Weight of fitting vs. smoothing in object function for Bernstein smoothing

H Command, Namelist /HISTRY/

Variable	Default Value	Default Unit	Description
NPLT(1:2,1:50)	0^a	-	Index of element in history records; NPLT(1,I) is independent variable, NPLT(2,I) is dependent variable
			Real records:
			1 – Time
			2 – PHIS, the synchronous phase on same scale as ϑ^b
			3 – PDOT, dp_s/dt
			4 – THBAR, the mean value of θ for the distribution
			5 – EBAR, the average energy of the distribution
			6 – THRMS, the rms spread in θ of the particles
			7 – ERMS, the rms energy spread
			8 – ES, the synchronous energy
			9 – E0, the energy on the reference orbit
			10 – ES-E0
			11 – THREF, the azimuth of a particle tracked from (0,ES)
			12 – EREF, the energy of a particle tracked from (0,ES)
			13 – EPSILON, the emittance^c
			14 – NUS, the synchrotron frequency
			15 – SBCKT, the RF "bucket" area
			16 – HBCKT, the RF "bucket" height
			17 – ETA, $\gamma_T^{-2} - \gamma^{-2}$
			18 – RSCALE; see **T** command
			19 – TAU, synchronous revolution period
			20 – PSIADD, phase feedback; see **L** command
			21 – DAMPL, voltage feedback factor; see **L** command
			22 – DELR, synchronous orbit radius - reference orbit radius
			Integer records:
			31 – TURN NUMBER
			32 – KNTSC, number of particles in $\varepsilon_\ell \neq 0$ partition of distribution
			Array records:
			51- 60 – SPARE(1-10)
			101-110 – EV(1-10)
			111-120 – PSI(1-10)
			121-130 – FREQ(I)-FRI(I), Change in frequency of source I
			201-250 – FAMPL(1-50), Fourier amplitudes; see **F** command
			251-300 – FAZE(1-50), Fourier phases; see **F** command
NWRT(1:2,1:50)	0^a	-	Indices of elements in history records; works like NPLT but writes records to FORTRAN UNIT 19
IMETA	0	-	Directs graphic output
			< 0 - Output written to HIGZ metafile
			0 - Output to device selected by IDEV
			> 0 - Output to both metafile and selected IDEV
XCRNR	135.	0.001	Fraction of full width of plot frame between lefthand edge and left side of plot
YCRNR	69.0	0.001	Fraction of full height of plot frame between bottom edge and bottom of plot
XAXISL	750.	0.001	Plot width as a fraction of frame width
YAXISL	475.	0.001	Plot height as a fraction frame height

^{a}The default value of 0 indicates to SUBROUTINE HISTORY that all of the desired history plots have been generated, so only the first set of consecutive nonzero entries to array NPLT or NWRT will select data.

^{b}See **A** command description.

^{c}EPSILON = ANORM $\tau(\sqrt{\sum \theta_i^2 \sum E_i^2 - (\sum \theta_i E_i)^2})/N$ eVs; see text for ANORM.

K Command, Namelist /KUTS/

Variable	Default Value	Default Unit	Description
KUT	0	-	Cut the last KUT particles from the distribution
KNTSET	0	-	Reset particle count
			$< 0 \Longrightarrow$ KNTSC $\longrightarrow$ 0; removes bunch-type distribution
			$> 0 \Longrightarrow$ KOUNT $\longrightarrow$ 0; removes entire distribution
K1	0	-	Starting point for a partial removal of the distribution
K2	^{a}	-	End point for partial removal of distribution
KLASS	0	-	Selects a partition of the distribution to be removed^b

^{a}Defaults to KOUNT

^{b}See PARTION in **P** command

L Command, Namelist /LLRF/

Variable	Default Value	Unit	Description
PHFBON	F	-	Activates phase feedback
NTUAVG	1	-	The number of past turns to average in computing the feedback; the default $NTUAVG = 1$ represents infinite-bandwidth feedback
NTURES	1	-	The number of turns for the feedback to respond; the present signal is compared to the signal of NTURES turns ago.
ITFB	0	-	The form of phase feedback: 0 – Critical damping 1 – Critical damping; gain reduced proportionally to ν^2 within non-adiabatic interval 2 – Fixed 3 – Fixed; gain reduced proportionally to ν^2 within non-adiabatic interval
FBFACT	1.0	-	The gain applied to the phase feedback
USEWT	F	-	Applies weight function W to phase signal over NTUAVG turns
W(1:NTUAVG)	0.0	-	Weight function multiplying phase signal
DLIMIT	5.7296	deg	The upper limit on the magnitude of the phase feedback on a given turn
VFBON	F	-	Activates voltage feedback
VFBFCTR	1.0	-	The gain applied to the voltage feedback
VLIMIT	.1	-	Limit on the fraction of voltage feedback (DAMPL)
FILBWD	'DUMMY'	-	Name of time/bunch length curve for voltage feedback (see text)
ETAJMP	0.0	-	The value of η^a at which to jump the phase of the RF
BLFBON	F	-	Activates fundamental beam loading feedback correction
BLFFON	F	-	Activates fundamental beam loading feed-forward correction
ICAV	1	-	Index of the rf system to use in beam loading calculation
RSHUNT	-	ohm	Loaded shunt resistance of cavity(ies) for beam loading calculation
QCAV	-	-	Loaded Q of cavity(ies) for beam loading calculation
DTUNE	0.0	-	$\Delta f / f$ detuning of cavity(ies) in beam loading calculation
ENQ	2.D10	-	Number of beam particles represented by entire macroparticle distribution [b]
NHALF	-	-	Number of bins spanning half a bucket in the longitudinal charge distribution
GAIN	0.0	-	Loop gain for feedback correction of beam loading
DELAY	-	s	Loop delay in feedback correction to beam loading
FTOP	-	Hz	Upper cutoff for beam loading correction feedback loop
WDTHFCTR	1.0	-	Ratio of feed-forward current pulse widths to bunch rms widths
CHGFCTR	1.0	-	Ratio of feed-forward current pulse charges to bunch charges

[a] $\eta = \gamma_T^{-2} - \gamma^{-2}$
[b] Can be read with **L** or **B**

M Command, Namelist /MRANGE/

Variable	Default Value	Unit	Description
IPU	.TRUE.	-	Set display to beam current
FASPEC	.FALSE.	-	Set display to fourier amplitude spectrum
FPSPEC	.FALSE.	-	Set display to fourier power spectrum
TMBEGIN	0.0	s	Time at which to start saving mountain range data
TMEND	0.0	s	Time after which to stop saving mountain range data
MRMPLOT	1	turn	Turn interval at which to record mountain range data
MRNBIN	100	-	Number of bins in plot range MRBMAX - MRBMIN
MRBMIN	[a]		Minimum ordinate for mountain range
MRBMAX	[b]		Maximum ordinate for mountain range

[a] Defaults to $-180°/$FRAC for current pickup data and to one for fourier amplitude or power spectrum.
[b] Defaults to $180°/$FRAC for current pickup data and NBINFFT/2 for fourier amplitude or power spectrum.

<table>
<tr><td colspan="4" align="center">N Command, Namelist /MRPLOT/</td></tr>
<tr><td rowspan="2">Variable</td><td colspan="2" align="center">Default</td><td rowspan="2" align="center">Description</td></tr>
<tr><td>Value</td><td>Unit</td></tr>
<tr><td>MRPMIN</td><td>0.0[a]</td><td>-</td><td>Minimum abscissa value for mountain range plot</td></tr>
<tr><td>MRPMAX</td><td>0.0[a]</td><td>-</td><td>Maximum abscissa value for mountain range plot</td></tr>
<tr><td>NTRACE</td><td>100</td><td>-</td><td>Number of traces on page</td></tr>
<tr><td>NSKIP</td><td>0</td><td>-</td><td>Number of records to be skipped between each trace</td></tr>
<tr><td>TOPTOB</td><td>0.7</td><td>-</td><td>The fraction of the vertical range over which NTRACE traces are to be plotted (approximate if TBASE=T)</td></tr>
<tr><td>SCALE</td><td>0.3</td><td>-</td><td>The height of the first trace, in units in which the entire vertical range of the plot is 1.0</td></tr>
<tr><td>MSTART</td><td>0[b]</td><td>-</td><td>Turn number at which to start plots</td></tr>
<tr><td>MSTOP</td><td>0[b]</td><td>-</td><td>Turn number at which to stop plots</td></tr>
<tr><td>TMSTART</td><td>0.0[b]</td><td>s</td><td>Time at which to start plots</td></tr>
<tr><td>TMSTOP</td><td>0.0[b]</td><td>s</td><td>Time at which to stop plots</td></tr>
<tr><td>TBASE</td><td>F</td><td>-</td><td>Switch causing plot trace separation to be proportional to time</td></tr>
<tr><td>NRNBIN</td><td>100</td><td>-</td><td>Number of points to plot on a trace</td></tr>
<tr><td>SMOOTH</td><td>0</td><td>-</td><td>Smoothing option
-1 — 1-2-1 averaging of adjacent bins
0 — No smoothing
1 — Bernstein polynomial smoothing</td></tr>
<tr><td>OBJWGT</td><td>0.05</td><td>-</td><td>Weight of fitting term of object function w/ smoothing term for polynomial smoothing</td></tr>
<tr><td>ITNO</td><td>1</td><td>-</td><td>Number of iterations for either 1-2-1 or Bernstein smoothing</td></tr>
<tr><td>LIM</td><td>F</td><td>-</td><td>Switch for plotting dotted lines connecting left-most and rightmost non-zero points of consecutive traces[c]</td></tr>
<tr><td>IDEV</td><td>1</td><td>-</td><td>Output device</td></tr>
<tr><td>IMETA</td><td>0</td><td>-</td><td>Directs graphic output
< 0 - Output written to HIGZ metafile
0 - Output to device selected by IDEV
> 0 - Output to both metafile and selected IDEV</td></tr>
<tr><td>XCRNR</td><td>135.</td><td>0.001</td><td>Fraction of full width of plot frame between lefthand edge and left side of plot</td></tr>
<tr><td>YCRNR</td><td>69.0</td><td>0.001</td><td>Fraction of full height of plot frame between bottom edge and bottom of plot</td></tr>
<tr><td>XAXISL</td><td>750.</td><td>0.001</td><td>Plot width as a fraction of frame width</td></tr>
<tr><td>YAXISL</td><td>475.</td><td>0.001</td><td>Plot height as a fraction frame height</td></tr>
</table>

[a]Defaults of 0.0 for MRPMIN and MRPMAX imply data is to be plotted over its entire range; abscissa units are degrees for IPU=T and harmonic number for FAxSPEC=T.

[b]The defaults of 0 for MSTART and MSTOP, or 0.0 for TMSTART and TMSTOP, imply that all mountain range records are to be plotted

[c]SMOOTH = 1 also required.

O Command, Namelist /GRAPH/

Variable	Default Value	Default Unit	Description
MPLOT	1000	turn	Output every MPLOT turns
IDEV	1	-	Virtual device number for graphical output
POSTP	F	-	Write all data in COMMON blocks to unit 18; do not call plotting routine.
TITL	exists	-	String of up to 50 characters for plot headings
PLTSW			Select plot options:
(1)	T	-	Draw phase space plot
(2)	T	-	Plot phase space points (different symbol for each class)
(3)	F	-	Interconnect points within each class
(4)	F	-	Draw lines at centroid and $\pm\sigma$
(5)	F	-	Draw voltage waveform
(6)	F	-	Set plot boundaries to turning points of contour
(7)	F	-	Suppress captions, axis labels, etc.
(8)	T	-	Plot θ histogram
(9)	F	-	Set θ histogram limits to turning points of contour
(10)	T	-	Plot E histogram
(11)	F	-	Set E histogram limits to turning points of contour
(12)	F	-	Plot fourier amplitudes
(13)	F	-	Include phases in plot of fourier spectrum
(15)	F	-	Plot the real and imaginary impedance from the input
(17)	T	-	Start bucket contour at unstable fixed point [a]
(18)	T	-	Start bucket contour above stable fixed point at $E = E_s + H_{bckt}$
(19)	F	-	Plot flow line points (different symbol for each class)
(20)	F	-	Interconnect flow line points within each class
(21)	F	-	Bunch perf. cond. wall voltage *vs.* ϑ
(22)	F	-	Include distribution in previous plot
(23)	F	-	Wake field (time domain) voltage *vs.* ϑ
(24)	F	-	Include distribution in previous plot
(25)	F	-	Frequency domain voltage *vs.* ϑ (inc. perf. cond. wall if FDSCON = T)
(26)	F	-	Include distribution in previous plot
(27)	F	-	Voltage from special time domain calculation for resonators em *vs.* ϑ
(28)	F	-	Include distribution in previous plot
(29)	F	-	Create don't-plot area in phase plane defined by four points [b]
(30)	F	-	Create don't-plot area in phase plane defined by matched contour [c]
(31)	F	-	PLTSW(31)=T defines don't-plot area *outside* closed figure
(32)	F	-	Net beam loading voltage *vs.* ϑ.
(33)	F	-	Plot current (net of beam, feed-forward, & feedback) driving cavity
NPJMP	1	-	In phase space plot, plot only every NPJMPth point
KLPLOT	0	-	Select classes in phase space plot and projections (see Sec. 2.2.3) 0 — All classes plotted $1 \leq KLPLOT \leq KLASSES$ — Plot class KLPLOT only
IRF	1	-	Selects voltage source for contour plotting: < 0 — No contour plotted 0–All active (NRF) sources 1-10 — Source IRF ($1 \leq IRF \leq NRF$)
ICONTUR	1	-	Select the type of reference contour to plot on phase space plot 0 — No contour 1 — Bucket contour 2 — Contour of initial bunch area SBNCH 3 — Contour of the specified area REFAREA 4 — Contour containing 95% of the particles 5 — Flow lines chosen by LINES, ELMIN, and ELMAX
REFAREA	0.1	eVs	Area of reference contour for ICONTUR = 3
LINES	1	-	Number of flow lines for ICONTUR = 5
ELMIN	0.	MeV	Energy above E_s for first flow line [d]
ELMAX	1.	MeV	Energy of top flow line
THPMIN	0.0[e]	deg	Lower θ limit for phase space plot
THPMAX	0.0[b]	deg	Upper θ limit for phase space plot
DEPMIN	0.0[f]	MeV	Lower E limit for phase space plot
DEPMAX	0.0[e]	MeV	Upper E limit for phase space plot
IEREF	1	-	Determines energy origin for phase space: 1 — E0, the reference energy (often = ES) 2 — ES, the synchronous energy 3 — EBAR, the average particle energy 4 — EREF, the reference particle energy [g]

[a] At least one of PLTSW(17) or PLTSW(18) must be true for ICONTUR=1 else program sets both .TRUE.

[b] see definitions of THEXCMI, THEXCPL,DEEXCMI,DEEXCPL

[c] see definition of SEXCL

[d] Program will modify ELMIN < 0.

[e] THPMIN and THPMAX both 0.0 results in a plotting range $-180°/\text{FRAC} \leq \theta \leq 180°/\text{FRAC}$.

[f] DEPMIN and DEPMAX both 0.0 results in a plotting range approximately the range of particle energies

[g] A particle which ESME tracks from the origin (0,ES) as a reference.

O Command, Namelist /GRAPH/, continued			
	Default		
Variable	Value	Unit	Description
NBINTH	50	-	The number of bins for the θ histogram
THBMIN	0.0^a	deg	Lower limit for θ histogram
THBMAX	0.0^e	deg	Upper limit for θ histogram
NBINE	50	-	The number of bins for the E histogram
EBMIN	0.0^b	MeV	Lower limit for E histogram
EBMAX	0.0^f	MeV	Upper limit for E histogram
IFBMIN	1	-	Lower limit for FFT plot - If specified, specify IFBMAX also
IFBMAX	0^c	-	Upper limit for FFT plot
CVBMIN	0.0^d	deg	Lower θ limit for all collective voltage plots
CVBMAX	0.0^h	deg	Upper limit for all collective voltage plots
THEXCMI	0.	deg	Lower theta value defining rectangular don't plot area
THEXCPL	0.	deg	Upper theta value defining rectangular don't plot area
DEEXCMI	0.	MeV	Lower energy value defining rectangular don't plot area
DEEXCPL	0.	MeV	Upper energy value defining rectangular don't plot area
SEXCL	0.	eVs	Area of matched contour defining a don't-plot area
DTHCURV	0.0	deg	Amount by which contour will be moved in θ direction
DECURV	0.0	MeV	Amount by which contour will be moved in E direction
DELCON	.01	-	Determine bucket to precision DELCON*360° w/ RF
KNTLIM	500000	-	Number of iterations of difference equation which will be attempted to close contour
IMETA	0	-	Directs graphic output $<$ 0 - Output written to HIGZ metafile 0 - Output to device selected by IDEV $>$ 0 - Output to both metafile and selected IDEV
XCRNR	135.	0.001	Fraction of full width of plot frame between lefthand edge and left side of plot
YCRNR	69.0	0.001	Fraction of full height of plot frame between bottom edge and bottom of plot
XAXISL	750.	0.001	Plot width as a fraction of frame width
YAXISL	475.	0.001	Plot height as a fraction frame height

[a]Limits of 0.0 for both THBMIN and THBMAX result in the plot range being the same as for the phase space plot.

[b]Limits of 0.0 for both EBMIN and EBMAX result in the plot range being the same as for the phase space plot.

[c]IFBMAX = 0 results in the upper limit being the greatest Fourier harmonic computed.

[d]Limits of 0.0 for both CVBMIN and CVBMAX result in the range for the plot being $\pm180°/\text{FRAC}$.

P Command, Namelist /POPL8/

Variable	Default Value	Default Unit	Description
KIND	1	-	Chooses the type of distribution to be generated: 1–Rectangular outline, NTH by NE points, limited by THMIN, THMAX, REMIN, REMAX 2–Uniform rectangular grid NTH by NE, limits as in KIND = 1 3–Random uniform distribution of NPOINT points within rectangular limits as in KIND = 1 4–Random uniform in θ, limits THMIN, THMAX; Gaussian in E, REMIN, REMAX = $\pm 2\sigma$, NPOINT points 5–Gaussian in θ, THMIN,THMAX = $\pm 2\sigma$; random uniform in E, limits REMIN, REMAX, NPOINT points 6–Rectangular grid, regular in θ, Gaussian in E, NTH by NE points .... The remaining distribution types, except for 11, 12, 15, and 20 are *matched*, the distribution limited by a contour of SBNCH eVs. .... 7–Bunch outline of NPOINT particles 8–Regular grid of approximately NTH by NE particles 9–Random uniform bunch of NPOINT particles within contour 10–Bi-Gaussian distribution of NPOINT particles, 95% within contour 11–NPOINT uniformly spaced particles on flow lines just above and below bucket boundary 12–NPOINT particles, random uniform in E, limits REMIN,REMAX, parabolic in θ, limits THMIN,THMAX 13–Parabolic distribution of NPOINT particles 14–Elliptical distribution of NPOINT particles 15–NPOINT particles, random uniform in θ, limits THMIN,THMAX; parabolic in E, limits at REMIN,REMAX 16–NPOINT particles within matched contour, low-noise uniform quasi-random 17–NPOINT particles within matched contour, low-noise gaussian quasi-random 18–NPOINT particles in matched contour, low-noise uniform pseudo-random (Sobel sequence) 19–NPOINT particles in matched contour, low-noise gaussian pseudo-random (Sobel sequence) 20 –NPOINT particles, pseudo-random within limits REMIN,REMAX, THMIN,THMAX
NPOINT	1	-	Number of particles generated for all distributions except KIND = 1, 2, 6, 8, in which NTH and NE are used
NTH	2	-	Number of grid points in θ direction
NE	2	-	Number of grid points in E direction
THMIN	-90.0	deg	Lower θ limit on rectangular distributions
THMAX	90.0	deg	Upper θ limit on rectangular distributions
REMIN	None	MeV	Lower energy limit on rectangular distributions; relative to the synchronous energy, ES
REMAX	None	MeV	Upper energy limit on rectangular distributions
SBNCH	0.1	eVs	Area within limiting contour
IPOP	1	-	Specifies RF source for generating bunch limiting contour: 0–All active (NRF) sources I–Source I ($1 \leq I \leq$ NRF)
THOFF	0.0	deg	Amount to displace distribution generated in current call to POPUL8 in θ direction
EOFF	0.0	MeV	Amount to displace currently generated distribution in E direction
THTRAN	0.0	deg	Amount to displace all particles (generated in this and previous calls to POPUL8) in θ direction
ETRAN	0.0	MeV	Amount to displace all particles in E direction
WINJ	0.0	MeV	Injection energy; when WINJ and PINJ both have default value of 0.0 injection is at energy $E_\circ$ of central orbit
PINJ	0.0	MeV/c	Injection momentum; alternate for WINJ above (*q. v.*)
PARTION	F	-	Partition distribution into separate classes;[a] each separate use of the **P** command with PARTION = T introduces a new partition
RENORM	F	-	Calculate ANORM for matched bunch so that EPSILON $\equiv$ SBNCH. RENORM is reset to .FALSE. at each use.

[a]Different classes of particles may be plotted with distinct symbols.

	R Command, Namelist /RING/		

Variable	Default Value	Unit	Description
REQ	None	m	The reference radius for the central orbit
GAMTSQ	None	-	Square of transition γ (< 0 acceptable)
ALPHA1	0.0	-	Coefficient of $(\Delta p/p)^2$ in series expansion for length difference between particle trajectory and reference orbit
ALPHA2	0.0	-	Coefficient of $(\Delta p/p)^3$ in series for path length difference
ALPHA3	0.0	-	Coefficient of $(\Delta p/p)^4$ in series for path length difference
W0I	None	MeV	Kinetic energy on the central orbit at T = TI
W0F	0.0	MeV	Kinetic energy on the central orbit at T = TF (*see text*)
P0I	None	MeV/c	Momentum on the central orbit at T = TI
P0F	0.0	MeV/c	Momentum on the central orbit at T = TF (*see text*)
TI	0.0	s	Start time of magnetic field change
TF	0.0	s	End time of magnetic field change
TSTART	0.0	s	Time at which tracking begins
FRAC	1.	-	Determines azimuthal periodicity, calculation restricted to $-180^\circ/\text{FRAC} \leq \vartheta \leq 180^\circ/\text{FRAC}$
NCAV	1	-	Number of rf locations (default usually adequate)
PIPRAD	1.0	m	Radius of beam pipe for particle loss
EBDRY	F	-	Switch to set absorbing beam pipe walls at REQ $\pm$ PIPRAD
DES	0.0	MeV	Constant energy offset of synchronous orbit relative to reference orbit
KURVEB	1	-	Magnetic field ramp from W0I to W0F or P0I to P0F: 1 – Linear 2 – Increasing parabolic 3 – Biased sinusoidal (See text for definitions CRA, CRB, CRC, TR3.) 4 – Ramp table to be read from file given by value of FILRMP 5 – Parabolic with final slope W0FDOT 6 – Parabolic with final slope P0FDOT 7 – Cubic with zero initial slope and curvature 8 – Decreasing parabolic
CRA	0.0	-	Time shift of first harmonic of KURVEB=3 ramp
CRB	0.0	-	Relative amplitude of second harmonic in KURVEB=3
CRC	0.0	-	Time shift of second harmonic for KURVEB=3
TR3	0.0	s	Half-period of fundamental for KURVEB=3; TR3=0. $\Rightarrow$ half-period = TF-TI
W0IDOT	0.0	MeV/s	Slope of parabolic ramp at TI (KURVEB=5)
W0FDOT	0.0	MeV/s	Slope of parabolic ramp at TF
P0IDOT	0.0	MeV/c/s	Slope of parabolic ramp at TI (KURVEB=6)
P0FDOT	0.0	MeV/c/s	Slope of parabolic ramp at TF
JNRAMP	F	-	Establishes starting point of ramp as point at which program finds itself–for smoothly piecing ramp segments together
FILRMP	'DUMMY'	-	The file name (full path) of the ramp table file (see 3.2.3)
GMAJMP	F	-	Set γ_T-jump on (*See text for compatibility.*)
KINDG	1	-	Type of γ_T variation: [a] 1 – Linear ($\gamma_T = \text{GAMPAR}(1) + \text{GAMPAR}(2) * T$) 2 – Decreasing exponential ($\gamma_T = \text{GAMPAR}(1) + \text{GAMPAR}(3) * (1 - e^{-T/\text{GAMPAR}(2)})$)
GAMPAR(1:3)	0.0	-	Coefficients for γ_T variation (real γ_T only)
CHGNO	1.0	-	Charge of beam particle in units $|e|$
EM0CSQ	938.27231	MeV	Rest energy of beam particle

[a]T = 0 corresponding to time at which **R** command is invoked with GMAJMP = .TRUE.

T Command, Namelist /CYCLE/

Variable	Default Value	Unit	Description
TSTOP[a]	0.0	s	Time at which to stop tracking
TTRACK[b]	0.0	s	Duration of time to track
MSTEP	100	-	Number of tracking steps (minimum) per synchrotron period
RSCALE0	1.0	-	Ratio of time per iteration to circulation period τ [c] — See text!
LGRTHM	1	-	Select difference equations used in tracking 1–Complete kinematics, expand path length to maximum order using input coefficients ALPHAn[d] 2–Use the simplified difference equation $\vartheta_{i,n} = \frac{\tau_{s,n}}{\tau_{s,n-1}}\vartheta_{i,n-1} + 2\pi\eta\frac{\Delta p}{p}$
ITRAP(1:4)	0	-	Indicates a condition for which tracking should be interrupted before time indicated by TTRACK or TSTOP: 0–No trap 1–Trap on minimum bunch width 2–Trap on minimum bunch height 3–Trap for $\eta = $ ETATRP (tolerance $\Delta\eta/\eta = \pm.01$) 4–Trap for $|\phi_s| = $ PHISTRP (tolerance $\Delta\phi_s = \pm.005$) 5–Trap for $\eta > 0$ (transition crossing) 10 – 19–Call SUBROUTINE SHAZAM, enter at SHAZAM, SHAZAM1, SHAZAM2, ... *following* every iteration of the difference equations -1 – -10–Call SUBROUTINE SHAZAM, enter at SHAZAM, SHAZAM1, SHAZAM2, ... *before* every iteration of the difference equations
ETATRP	.001	-	For ITRAP = 3; tracking stopped when $\eta = $ ETATRP
PHISTRP	.95	-	For ITRAP = 4; tracking stopped when $|\sin\phi_s| = $ PHISTRP
MGRACE	0	-	Allow a "grace period" of MGRACE turns before trapping conditions are checked
HISTRY	F	-	Write history records to unit 9 during tracking[e]
HISTSIZ	10^4	-	Approximate number of history records for one execution of **T**
BBDRY	F	-	Remove particles tracked outside of region $-180°/\text{FRAC} \leq \theta \leq 180°/\text{FRAC}$

[a]TSTOP is set to 0.0 when tracking is completed, or interrupted by an ITRAP option.

[b]TSTOP takes precedence; if TSTOP=0.0, then TTRACK determines duration of tracking.

[c]Scaling with RSCALE0 > 1 limited by MSTEP

[d]See R command

[e]See Section 3.2

Y Command, Namelist /MEMORY/

Variable	Default Value	Unit	Description
KNPHASE	50001	-	The storage allocation for each phase coordinate — max. value for KOUNT +1
KMAXCVB	1000	-	The storage allocation for large time domain tables [a]
KMAXPTS	1000	-	Storage allocation for history plots

[a]KMAXCVB sets the maximum allowable value for input parameters NBINSC and NBRES.

Bibliography

[1] J. A. MacLachlan and Z. Nazario, "Scaling for faster macroparticle simulation in longitudinal multiparticle dynamics", FERMILAB-Pub-00/169-T (July 2000)

[2] *Design Report Tevatron I Project*, Fermi National Accelerator Laboratory (October 1984), Chapters 4–6

[3] J. E. Griffin, J. A. MacLachlan, A. G. Ruggiero, K. Takayama, "Time and Momentum Exchange for the Production and Collection of Intense Antiproton Beams at Fermilab", IEEE Trans. Nucl. Sci. **30**#4(1983) 2630–2632

[4] J. E. Griffin, J. A. MacLachlan, Z. B. Qian," RF Exercises Associated with Acceleration of Intense Antiproton Bunches at Fermilab", IEEE Trans. Nucl. Sci. **30**#4(1983) 2627–2629

[5] J. A. MacLachlan, "ESME: Longitudinal Phasespace Particle Tracking — Program Documentation", Fermilab TM-1274(May 1984), unpublished (largely obsolete)

[6] J. MacLachlan, "Particle Tracking in E-ϕ Space for Synchrotron Design & Diagnosis", Fermilab-CONF-92/333 (Nov. 92), presented at 12th Int'l Conf. on Appl. of Acc. in Res. and Ind., Denton TX, 4 Nov. 1992

[7] J. A. MacLachlan, "Longitudinal Phasespace Tracking with Spacecharge and Wall Coupling Impedance", Fermilab FN-446 (February 1987), unpublished.

[8] P. Lucas and J. MacLachlan, "Simulation of Spacecharge Effects and Transition Crossing in the Fermilab Booster", Proc. 1987 IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987) p1114.

[9] S. Stahl and C. Ankenbrandt, "Simulation of the Capture Process in the Fermilab Booster", Proc. 1987 IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987)p1117

[10] P. Lucas and Q. Kerns, "Simulation of a Programmed Frequency Shift Near Extraction from the Fermilab Booster", Proc. IEEE Particle Accelerator Conference held at Washington, D.C. (16–19 March 1987)p1108

[11] S. Stahl and S. A. Bogacz, "Coupled Bunch Instability in a Circular Accelerator and Possible Cures: Longitudinal Phasespace Simulation", Phys. Rev. **D37**#5(1 March 1988)1300–1306

[12] J. A. MacLachlan, "Fundamentals of Particle Tracking for the Longitudinal Projection of Beam Phasespace in Synchrotrons", Fermilab FN-481(15 April 88), unpublished.

[13] S. Stahl and S. A. Bogacz, "Simulation of Coupled Bunch Mode Growth Driven by a High-Q Resonator: A Transient Response Approach", Proc. 1989 IEEE Particle Accelerator Conference held at Chicago IL (20–23 March 1989)p1175.

[14] G. Sabbi,"Simulation of Single-Bunch Effects in LEP by Linear Expansion of the Distribution Moments", CERN SL/95-25 (AP), CERN (9 May 1995), unpublished

[15] The copyright for the pgplot plotting package is held by California Inst. of Technology. It is written in standard f77 and is freely available for non-commercial use.

[16] A. Van Ginneken, "Smoothing Algorithm for Histograms of One or More Dimensions", Nucl. Inst. and Meth. in Phys. Res. **A305** (1991) pp 453–461